Podstawy optymalnego projektowania konstrukcji

Optymalizacja funkcji bez ograniczeń

Marcin Rodak
marcin.rodak@put.poznan.pl

Zakład Wytrzymałości i Konstrukcji
Instytut Mechaniki Stosowanej
Politechnika Poznańska

wrzesień 2015

Spis treści

Sformułowanie problemu

Problem optymalizacji funkcji bez ograniczeń (ZBO) możemy sformułować następująco:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) \tag{1.1}$$

gdzie $f : \mathbb{R}^n \to \mathbb{R}$ oraz $\boldsymbol{x} \in \mathbb{R}^n$

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}.$$

Sformułowanie problemu

W rzeczywistych zadaniach konstrukcyjnych przy poszukiwaniu rozwiąznia optymalnego rzadko się zdarza problem bez ograniczeń. Jednak analiza tej klasy problemów ma znaczenie ze względu na to, że

- istnieją problemy, w których ograniczenia nie mają wpływu na osiągnięte minimum,
- niektóre metody minimalizacji funkcji z ograniczeniami wymagają użycia algorytmów znanych z optymalizacji funkcji bez ograniczeń.

Klasyfikacja metod optymalizacji funkcji bez ograniczeń

Klasyczne metody numeryczne poszukiwania minimum funkcji bez ograniczeń można podzielić na

1. metody statystyczne (przypadkowe),
2. metody deterministyczne (zdeterminowane)

Klasyczne metody deterministyczne polegają na przeszukiwaniu przestrzeni $\mathbb{R}^n$ wzdłuż prostych zwanych *kierunkami poszukiwań*.
Metody te można podzielić ze względu na

1. sposób tworzenia kierunków poszukiwań:
   - metody o modyfikowanej bazie lub *bezgradientowe*,
   - metody o modyfikowanym kierunku lub *gradientowe*,
2. sposób znajdowania kolejnego punktu wzdłuż kierunku poszukiwań:
   - metody *dyskretne* lub *poszukiwań prostych*,
   - metody *z minimalizacją* lub *kierunków poprawy*,

Klasyfikacja metod optymalizacji funkcji bez ograniczeń

Metody bezgradientowe

polegają na poszukiwaniu minimum funkcji celu w n niezależnych kierunkach, a następnie wykonywana jest modyfikacja bazy na podstawie informacji o zmianach wartości funkcji f.

Metody gradientowe

korzystają z informacji o wartości i kierunku gradientu funkcji celu i na tej podstawie na bieżąco ustalają kierunki poszukiwań.

Metody poszukiwań prostych

badają zachowanie funkcji celu jednym lub dwóch punktach leżących na danym kierunku poszukiwań odległość tych punktów od punktu początkowego jest ustalona na początku każdej iteracji.

Metody kierunków poprawy

polegają na minimalizcji funkcji celu w kierunku poszukiwań.

Metoda Hooka-Jeevesa – wstęp

Metoda Hooka-Jeevesa jest bezgradientową metodą poszukiwań prostych.

Algorytm oparty na metodzie HJ dzieli się na dwa etapy:

- próbny, który polega na badaniu lokalnego zachowania funkcji w niewielkim wybranym obszarze poprzez wykonywanie kroków próbnych wzdłuż wszystkich kierunków ortogonalnej bazy,
- roboczy, który polega na przejściu w sposób zdeterminowany do następnego obszaru, w którym wykonywany jest kolejny etap próbny.

Etap roboczy wykonywany jest w przypadku, gdy w poprzedzającym go etapie próbnym został wykonany co najmniej jeden krok pomyślny. W przeciwnym wypadku następuje powrót do poprzednio badanego obszaru, w którym wystąpił pomyślny krok i wykonywany jest etap próbny z mniejszą wartością kroku.
Jako kryterium zatrzymania algorytmu należy przyjąć warunek, aktualna długość kroku jest mniejsza niż zadana liczba.

Metoda Hooka-Jeevesa – algorytm

Algorytm

Dane wejściowe:

$\boldsymbol{x}_0$ – punkt startowy,

$\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_n$ – ortogonalna baza,

λ – długość kroku,

β – współczynnki korekcyjny zmniejszający λ, $0 < \beta < 1$,

ε – dopuszczalna minimalna długość kroku (kryterium stopu)

$\boldsymbol{x}_{B0}$ – w pierwszej iteracji podstaw $\boldsymbol{x}_{B0} := \boldsymbol{x}_0$,

Na początku działania algorytmu oblicz

$$f_0 = f_{B0} = f(\boldsymbol{x}_0)$$

oraz podstaw

$$i := 1.$$

Metoda Hooka-Jeevesa – algorytm

Algorytm

Etap próbny

1. W kierunku $\boldsymbol{s}_i$ wykonaj krok próbny i oblicz wartość funkcji celu
$$\boldsymbol{x}_i = \boldsymbol{x}_{i-1} + \lambda \boldsymbol{s}_i \qquad f_t = f(\boldsymbol{x}_i).$$
2. Jeśli $f_t < f_0$ to podstaw $f_0 := f_t$ i przejdź do kroku (6)
3. W kierunku $\boldsymbol{s}_i$ wykonaj krok próbny i oblicz wartość funkcji celu
$$\boldsymbol{x}_i = \boldsymbol{x}_{i-1} - \lambda \boldsymbol{s}_i \qquad f_t = f(\boldsymbol{x}_i).$$
4. Jeśli $f_t < f_0$ to podstaw $f_0 := f_t$ i przejdź do kroku (6)
5. Podstaw $\boldsymbol{x}_i := \boldsymbol{x}_{i-1}$
6. Jeśli $i < n$ to $i := i + 1$ i przejdź do kroku (1).
7. Jeśli $f_0 < f_{B0}$ to podstaw $\boldsymbol{x}_B := \boldsymbol{x}_n$ (punkt bazowy), $f_B := f_0$ i wykonaj etap roboczy.
8. Podstaw $\boldsymbol{x}_0 := \boldsymbol{x}_{B0}$ oraz $\lambda := \beta\lambda$ i przejdź do punktu (1).
9. Jeśli $\lambda < \varepsilon$ to zakończ przyjmując $\boldsymbol{x}^* = \boldsymbol{x}_{B0}$ oraz $f^* = f_{B0}$.
10. Podstaw $i := 1$ i przejdź do punktu (1).

Metoda Hooka-Jeevesa – algorytm

Algorytm

Etap roboczy

1. Wykonaj krok roboczy i oblicz wartość funkcji celu

$$\boldsymbol{x}_0 = \boldsymbol{x}_B + (\boldsymbol{x}_B - \boldsymbol{x}_{B0}) = 2\boldsymbol{x}_B - \boldsymbol{x}_{B0}$$
$$f_0 = f(\boldsymbol{x}_0).$$

2. Podstaw

$$\boldsymbol{x}_{B0} := \boldsymbol{x}_B,$$
$$f_{B0} := f_B.$$

3. Podstaw $i := 1$ i przejdź do punktu (1) etapu próbnego.

Kierunki sprzężone

Definicja

Kierunki $\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_r$, $\boldsymbol{d}_i \neq 0$, $r \leqslant n$ nazywamy *wzajemnie sprzężonymi* względem dodatnio określonej macierzy $\mathbf{A}$ o wymiarach $n \times n$, jeżeli

$$\boldsymbol{d}_i^{\mathrm{T}} \mathbf{A} \boldsymbol{d}_j = 0 \qquad \text{dla} \qquad i, j = 1, 2, \ldots, r, \quad i \neq j.$$

Uwaga

W ten sposób zdefiniowane kierunki są liniowo niezależne.

Kierunki sprzężone

Twierdzenie

Niech dana będzie funkcja kwadratowa n zmiennych

$$Q(\boldsymbol{x}) = c + \boldsymbol{b}^{\mathrm{T}}\boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^{\mathrm{T}} A\boldsymbol{x}$$

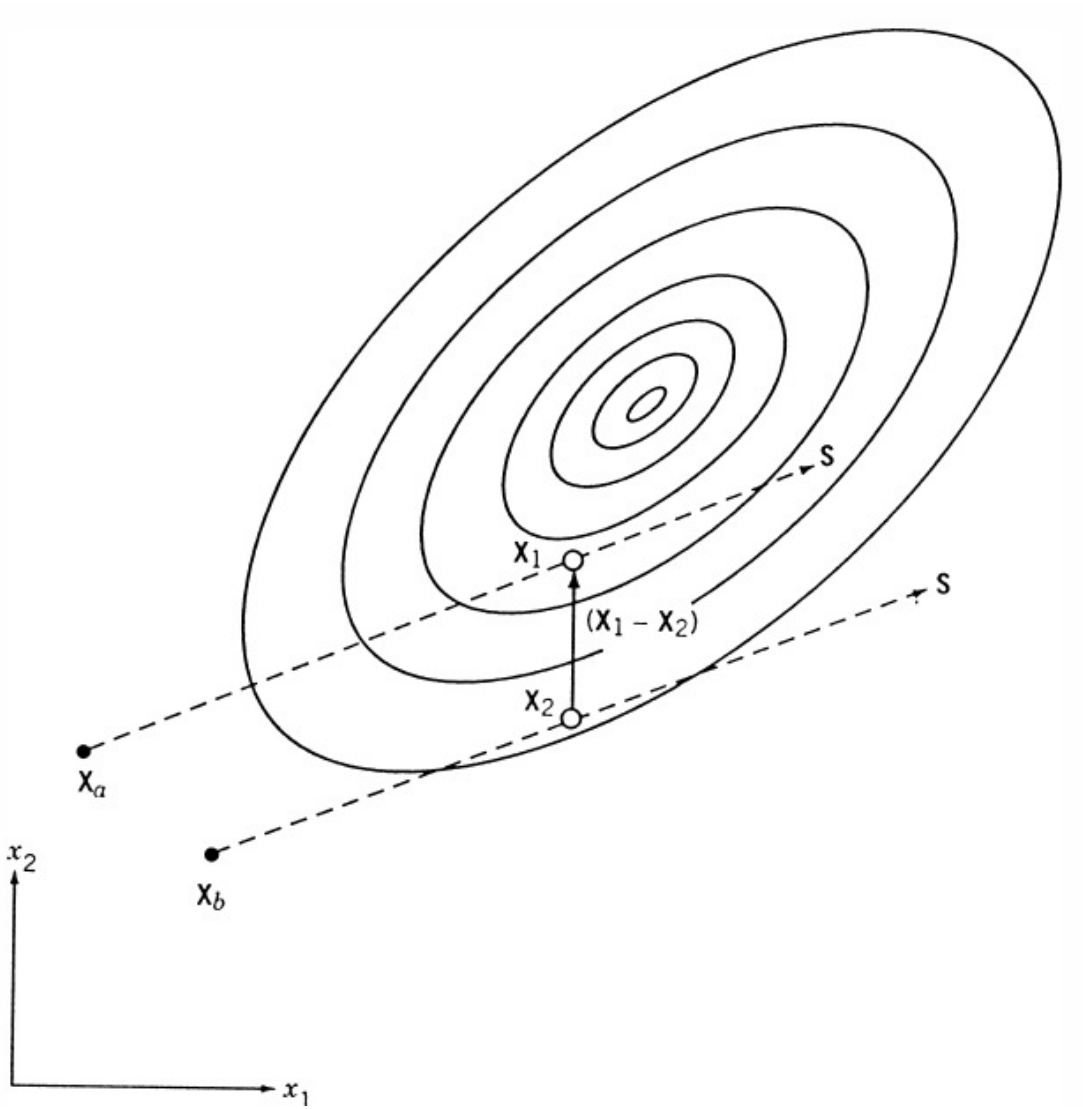
oraz niech $\boldsymbol{x}_1$ i $\boldsymbol{x}_2$ będą minimami funkcji $Q(\boldsymbol{x})$ na pewnych dwóch dowolnych równoległych do siebie rozmaitościach o wymiarach $k < n$.

$$Q(\boldsymbol{x}_1) = \min_{\boldsymbol{x}\in\mathbb{X}_1} Q(\boldsymbol{x}) \qquad\qquad Q(\boldsymbol{x}_2) = \min_{\boldsymbol{x}\in\mathbb{X}_2} Q(\boldsymbol{x}).$$

Wtedy kierunek $\boldsymbol{x}_2 - \boldsymbol{x}_1$ jest sprzężony względem macierzy A z dowolnym kierunkiem $\boldsymbol{d}$ równoległym do danych rozmaitości.

$$(\boldsymbol{x}_2 - \boldsymbol{x}_1)^{\mathrm{T}} A\boldsymbol{d} \quad \text{dla} \quad \boldsymbol{d}\|\mathbb{X}_1.$$

Kierunki sprzężone



Rysunek: Graficzne przedstawienie twierdzenia dla funkcji kwadratowej dwóch zmiennych

Kierunki sprzężone

Twierdzenie

Niech $\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_n$ będą wektorami wzajemnie sprzężonymi względem dodatnio określonej macierzy $\mathbf{A}$ oraz niech dana będzie funkcja kwadratowa n zmiennych

$$Q(\boldsymbol{x}) = c + \boldsymbol{b}^{\mathrm{T}}\boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^{\mathrm{T}} A \boldsymbol{x}.$$

Minimum formy $Q(\boldsymbol{x})$ może być wyznaczne w skończnej liczbie iteracji $\leqslant n$ w wyniku minimalizacji tej funkcji wzdłuż każdego kierunku $\boldsymbol{s}_i$ co najwyżej jeden raz.

Metoda Powell'a – wstęp

Metoda Powell'a jest bezgradientową metodą kierunków poprawy.

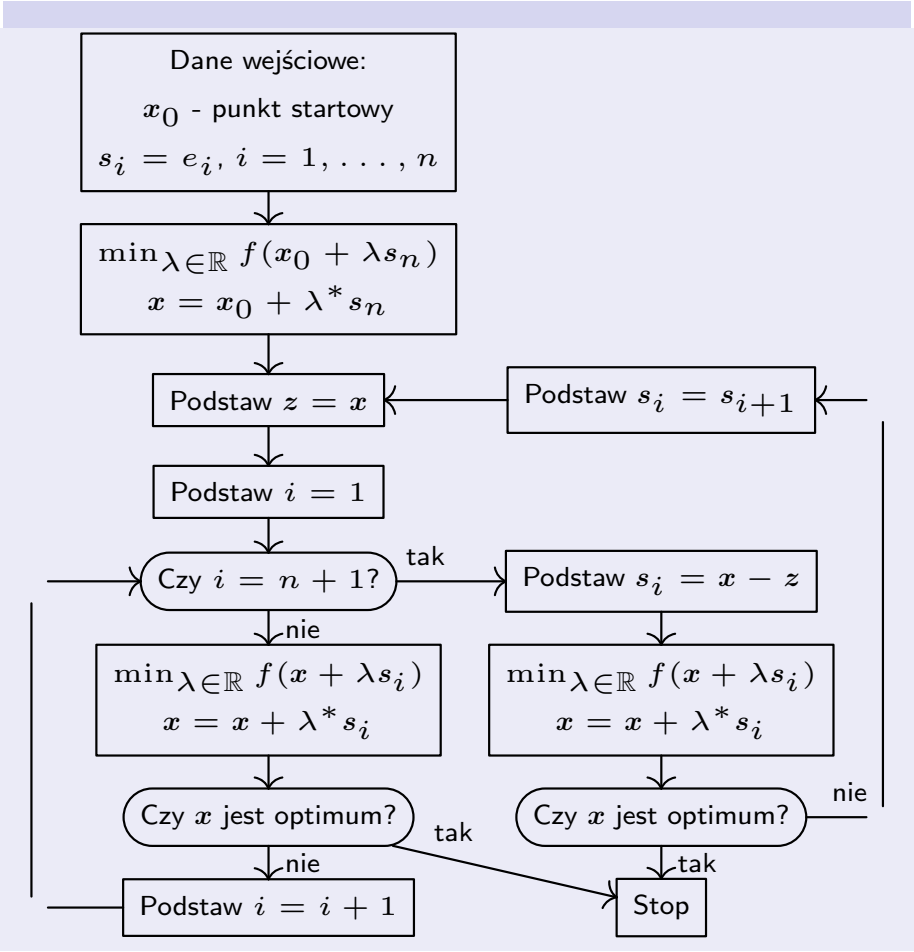
Metoda Powell'a można podzielić na etapy. W każdym z etapów

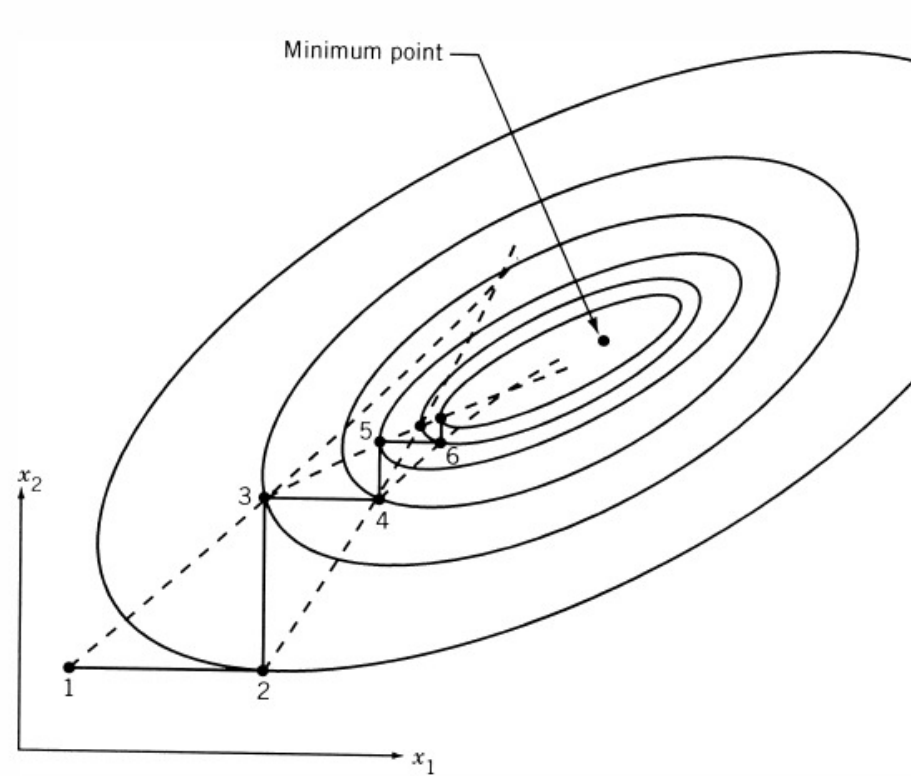
- przeprowadzana jest $n+1$-krotna minimalizacja funkcji celu w n kierunkach poszukiwań $\boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_n$

$$\min_{\lambda_i \in \mathbb{R}} f(\boldsymbol{x}_{i-1} + \lambda \boldsymbol{s}_i) = f(\boldsymbol{x}_{i-1} + \lambda^* \boldsymbol{s}_i),$$

- tworzony jest nowy kierunek poszukiwań $\boldsymbol{x}_{n+1} - \boldsymbol{x}_0$, który jest sprzężony z kierunkiem wzdłuż, którego funkcja celu była minimalizowana dwukrotnie w tym etapie,
- jeden z kierunków bazy zostaje zastąpiony nowym kierunkiem poszukiwań.

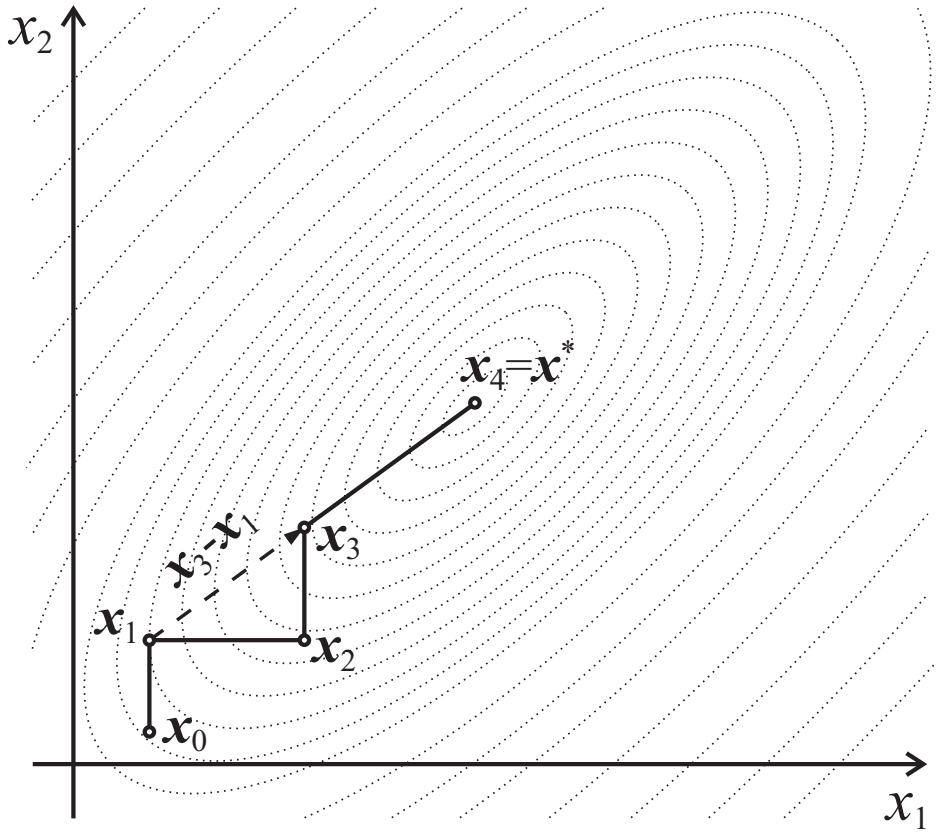
Metoda Powell'a – wstęp





Rysunek: Funkcja dwóch zmiennych

Metoda Powell'a –



Rysunek: Funkcja kwadratowa dwóch zmiennych

Metoda Powell'a – uwagi

Uwagi

Zbieżność metody Powell'a nie jest tak dobra przy praktycznym zastosowaniu jak wynikałoby to z przedstawionej teorii. Powodów tego jest kilka.

- Liczba etapów wynosi maksymalnie n tylko dla funkcji będącej formą kwadratową, natomiast w ogólnym przypadku, algorytm potrzebuje więcej etapów.
- Zbieżność jest kwadratowa przy założeniu, że znajdowane jest dokładne minimum funkcji w danym kierunku. W rzeczywistości jest to tylko punkt aproksymujący minimum, stąd obliczne kierunki nie są kierunkami wzajemnie sprzężonymi i dlatego metoda wymaga większej liczby iteracji.
- Prosta metoda Powell'a opisana wyżej może nie znaleźć minimum gdyż w trakcie obliczeń kierunki poszukiwań $\boldsymbol{s}_i$ mogą być od siebie liniowo zależne.

Metoda sympleksu Neldera–Meada – wstęp

Defininicja

Sympleksem n-wymiarowym nazywamy wielościan rozpięty na $n+1$ wierzchołkach, tzn.

$$\boldsymbol{x} = \sum_{i=1}^{n+1} \alpha_i \boldsymbol{x}_i,$$

gdzie

$$\sum_{i=1}^{n+1} \alpha_i = 1, \quad \text{i} \quad \alpha_i \geqslant 0.$$

Metoda sympleksu Neldera–Meada – wstęp

Metoda polega na porównywaniu wartości funkcji celu $f(\boldsymbol{x})$ w $n+1$ wierzchołkach sympleksu i iteracyjnemu poruszaniu sympleksu w kierunku optimum oraz zmianie jego wielkości.
Przemieszczenie i zmiama wymiarów sympleksu następuje w wyniku trzech operacji:

- odbicia,
- eksapnsji,
- kontrakcji.

Metoda sympleksu Neldera–Meada – wstęp

Oznaczenia

- punkt będący wierzchołkiem sympleksu, w którym funkcja celu osiąga maksymalną wartość oznaczamy $\boldsymbol{x}_h$, czyli

$$f_h = \max_{i=1,\ldots,n+1} f(\boldsymbol{x}_i),$$

- punkt będący wierzchołkiem sympleksu, w którym funkcja celu osiąga minimalną wartość oznaczamy $\boldsymbol{x}_l$, czyli

$$f_l = \max_{i=1,\ldots,n+1} f(\boldsymbol{x}_i),$$

- punkt będący środkiem n wierzchołkowych punktów sympleksu poza punktem $\boldsymbol{x}_h$ oznaczamy $\boldsymbol{x}_0$, czyli

$$\boldsymbol{x}_0 = \frac{1}{n} \sum_{\substack{i=1 \\ i \neq h}}^{n+1} \boldsymbol{x}_i,$$

Metoda sympleksu Neldera–Meada – wstęp

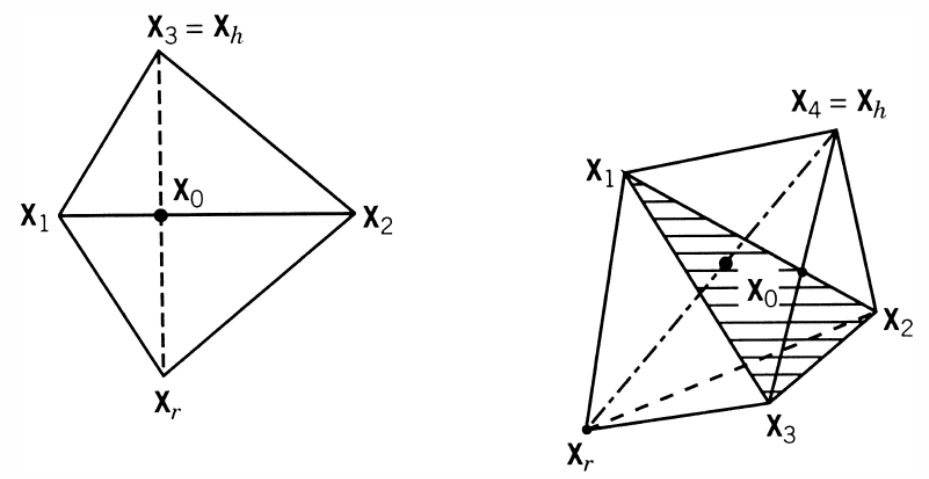
Operacja odbicia

Operacja odbicia polega na usunięciu wierzchołka sympleksu $\boldsymbol{x}_h$ i zastąpieniu go wierzchołkiem $\boldsymbol{x}_r$, który jest położony na linii łączcej punkty $\boldsymbol{x}_0$ i $\boldsymbol{x}_h$. Matematycznie operację tę zapisujemy:

$$\boldsymbol{x}_r = \boldsymbol{x}_0 + \alpha(\boldsymbol{x}_0 - \boldsymbol{x}_h) = (1+\alpha)\boldsymbol{x}_0 - \alpha\boldsymbol{x}_h,$$

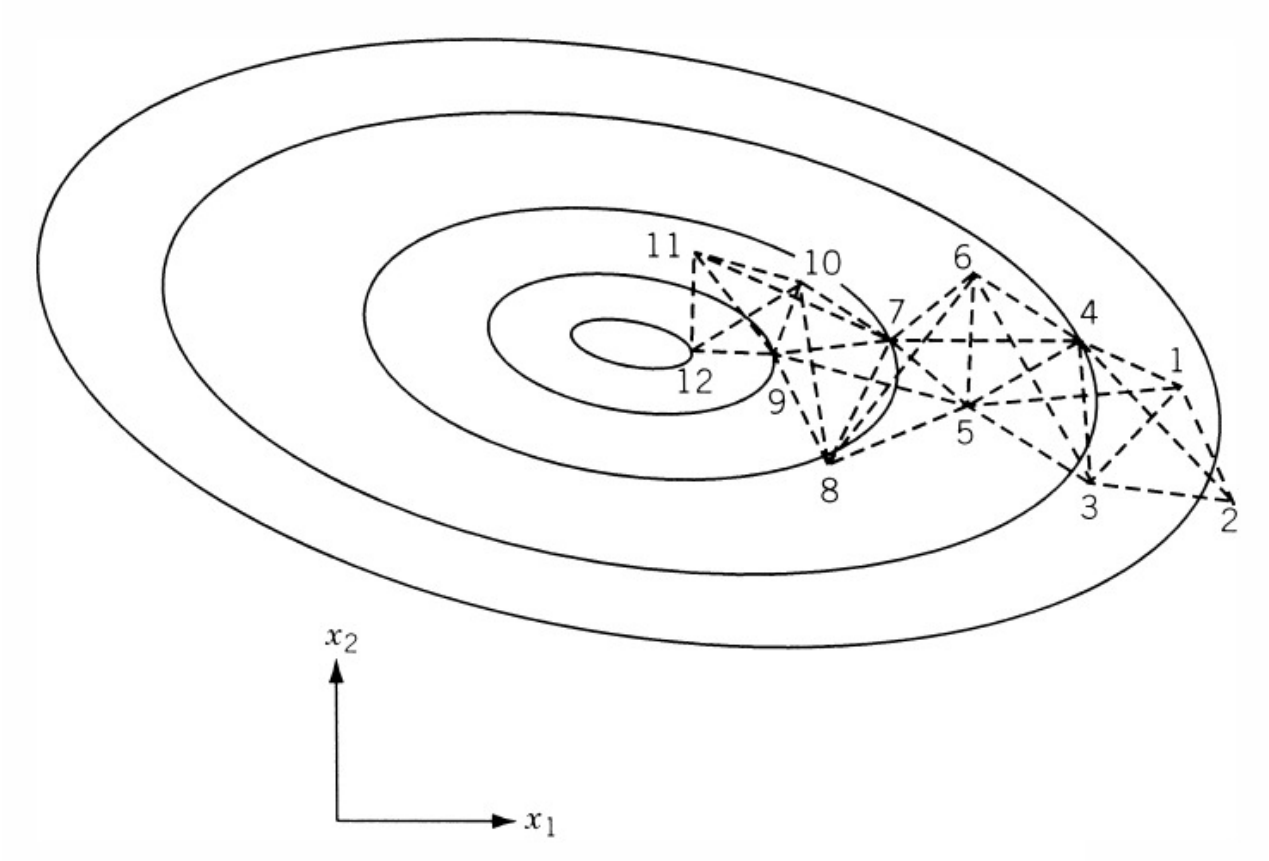
gdzie $\alpha > 0$ nazywamy współczynnikiem odbicia.

Wartość funkcji celu w tym punkcie oznaczmy $f_r = f(\boldsymbol{x}_r)$



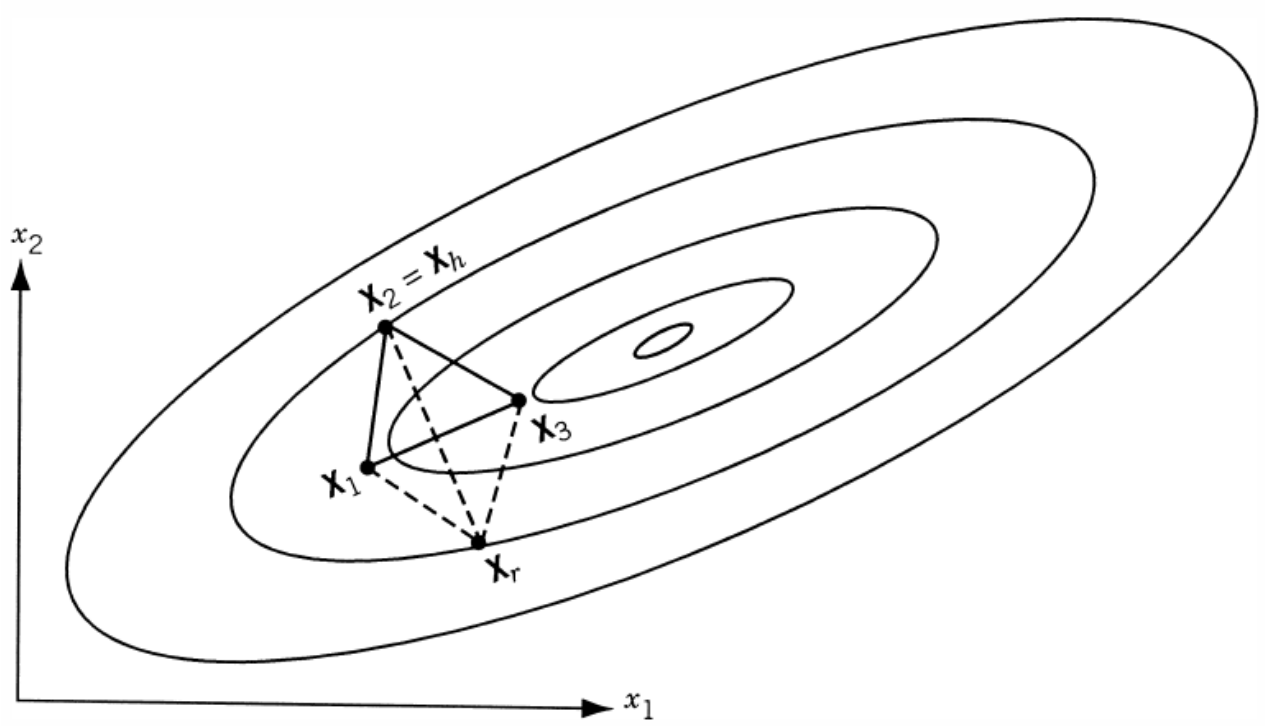
Rysunek: Operacja odbicia

Metoda sympleksu Neldera–Meada – wstęp



Rysunek: Przemieszczanie sie sympleksu w wyniku operacji odbicia

Metoda sympleksu Neldera–Meada – wstęp



Rysunek: Przykład rozbieżności algorytmu Neldera–Meada przy zastosowaniu tylko operacji odbicia

Metoda sympleksu Neldera–Meada – wstęp

Operacja ekspansji

Operacja ekspansji polega powiększeniu sympleksu poprzez przesunięcie jego wierzchołka $\boldsymbol{x}_r$. Matematycznie operację tę zapisujemy:

$$\boldsymbol{x}_e = \boldsymbol{x}_0 + \gamma(\boldsymbol{x}_r - \boldsymbol{x}_0) = \gamma\boldsymbol{x}_r + (1-\gamma)\boldsymbol{x}_0,$$

gdzie $\gamma > 1$ nazywamy współczynnikiem ekspansji.

Wartość funkcji celu w tym punkcie oznaczmy $f_e = f(\boldsymbol{x}_e)$

Metoda sympleksu Neldera–Meada – wstęp

Operacja kontrakcji

Operacja kontrakcji polega pomniejszeniu sympleksu poprzez przesunięcie jego wierzchołka $\boldsymbol{x}_h$. Matematycznie operację tę zapisujemy:

$$\boldsymbol{x}_c = \boldsymbol{x}_0 + \beta(\boldsymbol{x}_h - \boldsymbol{x}_0) = \beta\boldsymbol{x}_h + (1-\beta)\boldsymbol{x}_0,$$

gdzie $0 < \beta < 1$ nazywamy współczynnikiem kontrakcji.

Wartość funkcji celu w tym punkcie oznaczmy $f_c = f(\boldsymbol{x}_c)$

Metoda sympleksu Neldera–Meada

Przebieg algorytmu

Algorytm Neldera-Meada w każdym eatapie ma następujący:

1. wyznacz nowe indeksy l, h i oblicz $\boldsymbol{x}_0$,
2. wykonaj operację odbicia (oblicz $\boldsymbol{x}_r$, f_r),
3. jeżeli $f_r < f_l$ to wykonaj operację ekspansji (oblicz $\boldsymbol{x}_e$, f_e),
   - jeżeli $f_e < f_l$ to podstaw $\boldsymbol{x}_h := \boldsymbol{x}_e$ i rozpocznij nowy etap algorytmu,
   - jeżeli $f_e \geqslant f_l$ to podstaw $\boldsymbol{x}_h := \boldsymbol{x}_r$ i rozpocznij nowy etap algorytmu,
4. jeżeli istnieje $i = 1, 2 \ldots, n+1$ oraz $i \neq h$ takie, że $f_r < f_i$ to podstaw $\boldsymbol{x}_h := \boldsymbol{x}_r$ i rozpocznij nowy etap algorytmu
5. jeżeli $f_r \geqslant f_i$ dla $i = 1, 2 \ldots, n+1$, $i \neq h$ oraz $f_r < f_h$ to podstaw $\boldsymbol{x}_h := \boldsymbol{x}_r$,
6. wykonaj operację kontrakcji (oblicz $\boldsymbol{x}_c$, f_c),
7. jeżeli $f_c < f_h$ to podstaw $\boldsymbol{x}_h := \boldsymbol{x}_c$ i rozpocznij nowy etap algorytmu,
8. jeżeli $f_c \geqslant f_h$ to wykonaj redukcję całego sympleksu według wzoru $\boldsymbol{x}_i = \frac{1}{2}(\boldsymbol{x}_i + \boldsymbol{x}_l)$, a następnie rozpocznij nowy etap algorytmu.

Kierunki poprawy

Przez kierunek poprawy $\boldsymbol{s}$ w punkcie $\boldsymbol{x}$ możemy rozumieć taki kierunek poszukiwań, że w otoczeniu punktu $\boldsymbol{x}$ funkcja celu f maleje w kierunku $\boldsymbol{s}$, tzn.

$$\exists_{\delta>0} \forall_{0<t<\delta} \; f(\boldsymbol{x}+t\boldsymbol{s}) < f(\boldsymbol{x})$$

Twierdzenie

Jeśli funkcja $f : \mathbb{R}^n \to \mathbb{R}$ jest ciągła i różniczkowalna to wektor $\boldsymbol{s} \neq \boldsymbol{0}$ jest kierunkiem poprawy w punkcie $\boldsymbol{x}$ wtedy i tylko wtedy, gdy

$$\nabla f^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{s} < 0.$$

Dowód

Weźmy funkcję postaci

$$g(t) = f(\boldsymbol{x} + t\boldsymbol{s}).$$

Zauważmy, że nierówność $f(\boldsymbol{x} + t\boldsymbol{s}) < f(\boldsymbol{x})$ zachodzi dla pewnego $t > 0$, gdy $g(t) < g(0)$. Pochodna tej funkcji g równa się

$$\frac{\mathrm{d}g(t)}{\mathrm{d}t} = \sum_{i=1}^{n} \frac{\partial f(\boldsymbol{x} + t\boldsymbol{d})}{\partial x_i} \boldsymbol{s}_i$$

a w punkcie $t = 0$

$$\left.\frac{\mathrm{d}g(t)}{\mathrm{d}t}\right|_{t=0} = \sum_{i=1}^{n} \frac{\partial f(\boldsymbol{x})}{\partial x_i} \boldsymbol{s}_i = \nabla f^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{s}$$

Funkcja jednej zmiennej (ciągła i różniczkowalna) jest malejąca w dowolnym punkcie, gdy jej pochodna jest mniejsza od zera. Zatem

$$\nabla f^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{s} = \left.\frac{\mathrm{d}g(t)}{\mathrm{d}t}\right|_{t=0} < 0$$

□

Kierunki poprawy

Wniosek z dowodu twierdzenia

Poruszając się z punktu $\boldsymbol{x}$ w kierunku $-\nabla f(\boldsymbol{x})$ wartość funkcji celu zmniejsza się o największą wartość (kierunek największego spadku).

Weźmy dowolny wektor o długości 1, czyli $\|\boldsymbol{s}\| = 1$ wyrażenie $\nabla f^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{s}$ jest iloczynem skalarnym dwóch wektorów, czyli

$$\nabla f^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{s} = \left\|\nabla f^{\mathrm{T}}(\boldsymbol{x})\right\| \|\boldsymbol{s}\| \cos\theta = \left\|\nabla f^{\mathrm{T}}(\boldsymbol{x})\right\| \cos\theta,$$

gdzie θ to kąt pomiędzy $\nabla f(\boldsymbol{x})$ i $\boldsymbol{s}$. Wyrażenie to osiąga najmniejszą wartość, gdy kąt $\theta = \pi$. Zatem wektor $\boldsymbol{s}$ jest równoległy do wektora gradientu i przeciwnie skierowany

$$\boldsymbol{s} = -\frac{\nabla f(\boldsymbol{x})}{\|\nabla f(\boldsymbol{x})\|}$$

Metoda największego spadku – algorytm

Metoda największego spadku polega na obiearaniu w kolejnych iteracjach za kierunek poszukiwań wektora $-\nabla f(\boldsymbol{x})$. Jest to więc gradientow metoda kierunków poprawy.

Algorytm

1. Rozpocznij w wybranym punkcie startowym $\boldsymbol{x}_1$ oraz podstaw $i = 1$.
2. Oblicz $\nabla f_i = \nabla f(\boldsymbol{x}_i)$.
3. Znajdź optymalną wartość kroku λ_i^* oraz

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \lambda_i^* \nabla f_i,$$

4. Sprawdź czy $\boldsymbol{x}_{i+1}$ jest punktem optymalnym. Jeśli tak, to zakończ algorytm, w przeciwnym przypadku kontynuuj.
5. Podstaw $i = i + 1$ i przejdź do kroku (2).

Metoda największego spadku – uwagi

Uwagi

- kierunek nawiększego spadku/wzrostu jest własnością lokalną, a nie globalną (jest różny się w różnych punktach nawet jeśli jeden z tych punktów leży w linii największego spadku/wzrostu drugiego,
- podczas przebiegu algorytmu kierunki poszukiwań mają "zygzakowaty" charakter,
- trudność w dokładnym określeniu kierunku nawiększego spadku/wzrosu w miejscach gdzie funkcja celu ma charakter "wąskiej doliny".

Metoda Fletchera-Reevsa – wstęp

Metoda Fletchera-Reevsa różni się od metody największego spadku tym, że każdy kolejny kierunek poszukiwań obliczany jest następująco

$$\boldsymbol{s}_i = -\nabla f(\boldsymbol{x}_i) + \beta_i \boldsymbol{s}_{i-1}.$$

Współczynniki β_i są dobierane w ten sposób, aby przy minimalizacji funkcji kwadratowej

$$Q(\boldsymbol{x}) = c + \boldsymbol{b}^{\mathrm{T}}\boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\mathbf{A}\boldsymbol{x},$$

kolejne kierunki poszukiwań były wzajemnie sprzężone względem dodatnio określonej macierzy $\mathbf{A}$.
Prowadzi to do następującego wzoru

$$\beta_i = \frac{\nabla f^{\mathrm{T}}(\boldsymbol{x}_i)\nabla f(\boldsymbol{x}_i)}{\nabla f^{\mathrm{T}}(\boldsymbol{x}_{i-1})\nabla f(\boldsymbol{x}_{i-1})}$$

Metoda Fletchera-Reevsa

Przebieg algorytmu

Algorytm Fletchera-Reevsa ma następujący przebieg:

1. rozpocznij w wybranym punkcie startowym $\boldsymbol{x}_1$,
2. podstaw kierunek $\boldsymbol{s}_1 = -\nabla f(\boldsymbol{x}_1) = -\nabla f_1$,
3. podstaw $i := i + 1$,
4. oblicz $\boldsymbol{x}_i$ takie, że

$$f(\boldsymbol{x}_i) = f_i = \min_{\lambda \in \mathbb{R}} f(\boldsymbol{x}_{i-1} + \lambda \boldsymbol{s}_{i-1}),$$

5. sprawdź warunki końca działania algorytmu. Jeżeli są spełnione to zakończ działanie podstawiając $\boldsymbol{x}^* = \boldsymbol{x}_i$, $f(\boldsymbol{x}^*) = f(\boldsymbol{x}_i)$. W przeciwnym przypadku przejdź do natępnego punktu,
6. oblicz nowy kierunek poszukiwań według wzoru

$$\boldsymbol{s}_i = -\nabla f(\boldsymbol{x}_i) + \frac{\nabla f^{\mathrm{T}}(\boldsymbol{x}_i) \nabla f(\boldsymbol{x}_i)}{\nabla f^{\mathrm{T}}(\boldsymbol{x}_{i-1}) \nabla f(\boldsymbol{x}_{i-1})} \boldsymbol{s}_{i-1}$$

i przejdż do punktu (3).

Metoda Fletchera-Reevsa

Uwagi

- Algorytm Fletchera-Reevsa teoretycznie po n krokach powinien osiągnąć punkt optymalny, funkcja celu jest funkcją kwadratową. Zatem jest to algotytm kwadratowo zbieżny podobnie jak metoda Powell'a.
- W praktyce podczas przebiegu algorytm potrzebuje więcej iteracji do osiągnięcia optimum, szczególnie, gdy macierz $\mathbf{A}$ jest źle uwarunkowana. Wynika to z kumulacji błędów podczas obliczeń kolejnych kierunków poszukiwań. Są one obliczane na podstawie gradientu funkcji w punktach, które powinny być minimum funkcji w poprzednim kierunku, a w praktycznych obliczeniach jest to niemożliwe.

Metoda Newtona – wstęp

Metoda Newtona polega na aproksymacji funkcji celu w otoczeniu punktu $\boldsymbol{x}_i$ formą kwadratową postaci

$$Q(\boldsymbol{x}) = f_i + \nabla f_i^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{x}_i) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_i)^{\mathrm{T}}\mathbf{H}_i(\boldsymbol{x} - \boldsymbol{x}_i),$$

gdzie $\boldsymbol{x}_i$ jest przybliżeniem punktu optymalnego w i-tej iteracji, $f_i = f(\boldsymbol{x}_i)$, $\nabla f_i = \nabla f(\boldsymbol{x}_i)$ oraz $\mathbf{H}_i$ to hesjan funkcji f w punkcie $\boldsymbol{x} = \boldsymbol{x}_i$.

Forma kwadratowa ma minimum, gdy

$$\frac{\partial Q}{\partial x_j} = 0 \quad \text{dla} \quad j = 1, 2, \ldots, n.$$

Zatem

$$\nabla Q(\boldsymbol{x}) = \nabla f + \mathbf{H}_i(\boldsymbol{x} - \boldsymbol{x}_i) = 0.$$

Z tego równania można wyznaczyć punkt stacjonarny formy kwadratowej gdy macierz $\mathbf{H}_i$ jest nieosobliwa

$$\tilde{\boldsymbol{x}} = \boldsymbol{x}_i - \mathbf{H}_i^{-1}\nabla f_i.$$

Metoda Newtona

W metodzie Newtona w kolejnych iteracjach poczynając od punktu startowego $\boldsymbol{x}_1$ oblicza się punkty przybliżające szukane minimum według wzoru

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \mathbf{H}_i^{-1} \nabla f_i.$$

Funkcja celu f może nie być funkcją kwadratową dlatego metoda Newtona może zawodzić, tzn. być rozbieżna lub zbieżna do punktu siodłowego lub maksimum. W celu uniknięcia tego problemu kolejne punkty przybliżające optimum oblicza się następująco

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \lambda_i^* \mathbf{H}_i^{-1} \nabla f_i.$$

gdzie λ_i^* długością kroku minimalizującą funkcję celu w kierunku $-\mathbf{H}_i^{-1} \nabla f_i$.

Metoda Newtona

Uwagi

Metoda Newtona

- znajduje minimum funkcji kwadratowej w jednym kroku,
- wymaga magazynowania w pamięci komputera macierzy $\mathbf{H}_i$ wymiaru $n \times n$,
- wymaga trudnych, czasochłonnych lub niemożliwych obliczeń elementów macierzy $\mathbf{H}_i$,
- wymaga obliczenia odwrotności hesjanu,
- wymaga obliczenia wektora $\mathbf{H}_i^{-1} \nabla f_i$.

Ostatnie cztery cechy tej metody sprawiają, że jest ona niepraktyczna.

Metoda Davidona-Fletchera-Powella – wstęp

Metoda Davidona-Fletchera-Powella polega na wyznaczaniu kolejnych przybliżeń punktu, w którym funkcja celu osiąga minimum według wzoru

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \lambda_i^* \mathbf{B}_i \nabla f(\boldsymbol{x}_i),$$

gdzie λ_i^* długością kroku minimalizującą funkcję celu w kierunku $-\mathbf{B}_i \nabla f_i$, natomiast $\mathbf{B}_i$ aproksymuje odwrotność hesjanu $-\mathbf{H}_i^{-1}$ w punkcie $\boldsymbol{x} = \boldsymbol{x}_i$.

Metoda Davidona-Fletchera-Powella – wstęp

W metodzie tej w każdym kolejnym kroku macierz $\mathbf{B}$ jest aktualizowana

$$\mathbf{B}_{i+1} = \mathbf{B}_i + \Delta\mathbf{B}_i,$$

$$\Delta\mathbf{B}_i = \lambda_i^* \frac{\boldsymbol{s}_i \boldsymbol{s}_i^{\mathrm{T}}}{\boldsymbol{s}_i^{\mathrm{T}} \boldsymbol{g}_i} - \frac{(\mathbf{B}_i \boldsymbol{g}_i)(\mathbf{B}_i \boldsymbol{g}_i)^{\mathrm{T}}}{\boldsymbol{g}_i^{\mathrm{T}} \mathbf{B}_i \boldsymbol{g}_i},$$

$$\boldsymbol{g}_i = \nabla f(\boldsymbol{x}_{i+1}) - \nabla f(\boldsymbol{x}_i)$$

Metoda Davidona-Fletchera-Powella – wstęp

Algorytm

1. Rozpocznij w wybranym punkcie startowym x_1 oraz z dodatnio określoną macierzą aproksymującą odwrotność hesjanu funkcji f ($\mathbf{B}_1 = \mathbf{I}$), podstaw $i = 1$.
2. Oblicz $\nabla f_i = \nabla f(\boldsymbol{x}_i)$ oraz $\boldsymbol{s}_i = -\mathbf{B}_i \nabla f_i$.
3. Znajdź optymalną wartość kroku λ_i^* oraz
$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \lambda_i^* \boldsymbol{s}_i,$$
4. Sprawdź czy $\boldsymbol{x}_{i+1}$ jest punktem optymalnym. Jeśli tak, to zakończ algorytm, w przeciwnym przypadku kontynuuj.
5. Aktualizuj macierz
$$\mathbf{B}_{i+1} = \mathbf{B}_i + \Delta\mathbf{B}_i.$$
6. Podstaw $i = i + 1$ i przejdź do kroku (2).

Metoda Broydena-Fletchera-Goldfarba-Shanno – wstęp

Zasada działania metody BFGS jest taka sama jak metody DFP. Różnica polega na innej postaci macierzy aktualizującej macierz aproksymującą hesjan.

$$\Delta \mathbf{B}_i = \left(1 + \frac{\boldsymbol{g}_i^{\mathrm{T}} \mathbf{B}_i \boldsymbol{g}_i}{\boldsymbol{d}_i^{\mathrm{T}} \boldsymbol{g}_i}\right) \frac{\boldsymbol{d}_i \boldsymbol{d}_i^{\mathrm{T}}}{\boldsymbol{d}_i^{\mathrm{T}} \boldsymbol{g}_i} - \frac{\boldsymbol{d}_i \boldsymbol{g}_i^{\mathrm{T}} \mathbf{B}_i}{\boldsymbol{d}_i^{\mathrm{T}} \boldsymbol{g}_i} - \frac{\mathbf{B}_i \boldsymbol{g}_i \boldsymbol{d}_i^{\mathrm{T}}}{\boldsymbol{d}_i^{\mathrm{T}} \boldsymbol{g}_i},$$

gdzie

$$\begin{aligned} \boldsymbol{d}_i &= \boldsymbol{x}_{i+1} - \boldsymbol{x}_i = \lambda_i^* \boldsymbol{s}_i, \\ \boldsymbol{g}_i &= \nabla f_{i+1} - \nabla f_i = \nabla f(\boldsymbol{x}_{i+1}) - \nabla f(\boldsymbol{x}_i). \end{aligned}$$