Metody Obliczeniowe w Nauce i Technice

Minimalizacja funkcji

Marian Bubak, Katarzyna Rycerz

Department of Computer Science
AGH University of Science and Technology
Krakow, Poland
kzajac@agh.edu.pl
dice.cyfronet.pl

Contributors
Anna Bukowska
Yurii Vyzhha
Arkadiusz Placha

Wstęp

Motywacja

Szeroka klasa zagadnień – szukanie najmniejszej wartości przyjmowanej przez funkcję jednej lub wielu zmiennych.

Przykłady

- Produkcja - minimalizacja kosztów, przy zaspokojeniu popytu
- Zasoby ludzkie - problem przydziału godzin pracy
- Rolnictwo - dobór nawozów w celu maksymalizacji plonów
- Fizyka
- ...

Terminologia

Terminologia

- tradycyjnie: **minimalizacja**
- maksymalizacja ($F = -f(x)$)
- optymalizacja
- b. stary termin: programowanie (liniowe, nieliniowe, matematyczne)
- ekstremalizacja
- mathematical optimization (function minimization)

Zdefiniowanie zagadnienia

Definicja

Dane: funkcja $F : A \to \mathbb{R}, A \subseteq R^d$
Szukane: element $x_{min} \in A$ taki, że $\forall x \in A \; F(x_{min}) \leqslant F(x)$

Założenia

1. Dla funkcji F może nie być znany wzór analityczny.
2. Wartości x mogą być zawężone do pewnego ustalonego obszaru – constrained minization.
3. Mogą być dostępne $\frac{\partial F}{\partial x}$.
4. x_{min} jest wyliczane procedurą – wykorzystanie wielu wartości funkcji $f(x)$, aż do osiągnięcia minimum.

Zdefiniowanie zagadnienia

Kryteria wyboru najlepszej metody

Najlepsza metoda znajduje minimum (z zadaną tolerancją):

- po najmniejszej liczbie obliczeń wartości funkcji f
- po najmniejszej liczbie kroków procedury – niska złożoność obliczeniowa
- wymaga mało pamięci dodatkowej – niska złożoność pamięciowa

Gdzie szukać minimum globalnego funkcji?

$F(x)$ przyjmuje minimum w jednym z punktów:

1. p. stacjonarny – wszystkie $\frac{\partial F}{\partial x} = 0$
2. wierzchołek (cusp) – niektóre $\frac{\partial F}{\partial x}$ nie istnieją
3. edge point – na krawędzi obszaru

Gdy nie znamy funkcji analitycznie – musimy ograniczyć się do minimum lokalnego x_0 – *unconstrained local minimization*.

Minimum lokalne

$\forall x \in U,\ F(x) \geqslant F(x_0)$, gdzie U - otoczenie punktu x_0

Kształt funkcji $F(x)$ – 1-D

Założenie

$F(x)$ – ma sens fizyczny tj. w rozpatrywanym obszarze istnieją jej wszystkie pochodne.

Rozwijamy $F(x)$ wokół x_0 w **szereg Taylora (1-D)**:

$$F(x) = F(x_0) + \left.\frac{\partial F}{\partial x}\right|_{x_0} (x - x_0) + \left.\frac{1}{2}\frac{\partial^2 F}{\partial x^2}\right|_{x_0} (x - x_0)^2 + \dots$$

- Im mniejsza wartość $(x - x_0)$, tym mniej ważne człony wyższych rzędów.

Konkluzja: Dla małych kroków – przewidywania oparte na wyrazach niższego rzędu powinny być wystarczające.

Kształt funkcji $F(x)$ – n-D

Dla n-D: $\vec{x} = x = [x_1, x_2, ..., x_n]^T$

Rozwijamy funkcję F wokół $\vec{x_0}$:

$$F(\vec{x}) = \underbrace{F(\vec{x_0})}_{\text{stały}} + \vec{g}^T \cdot (\vec{x} - \vec{x_0}) + \frac{1}{2}(\vec{x} - \vec{x_0})^T \cdot H \cdot (\vec{x} - \vec{x_0}) + \dots$$

$$g_i = \left.\frac{\partial F}{\partial x_i}\right|_{\vec{x_0}} \text{ – gradient;} \quad H_{ij} = \left.\frac{\partial^2 F}{\partial x_i \partial x_j}\right|_{\vec{x_0}} \text{ – Hesjan;}$$

Kształt funkcji $F(x)$ – n-D

1. $\vec{g}^T \cdot (\vec{x} - \vec{x_0})$ – proporcjonalny do gradientu, wskazuje kierunek największego spadku funkcji.
   W pobliżu minimum: $\vec{g} \longrightarrow 0, to\ \vec{g} \cdot (\vec{x} - \vec{x_0}) \longrightarrow 0$ – człon liniowy, nie przepowiada minimum, nie można go użyć do określenia wielkości kroku.
2. $\frac{1}{2}((\vec{x} - \vec{x_0})^T) \cdot H \cdot (\vec{x} - \vec{x_0})$ – człon kwadratowy, najniższy człon przydatny do przewidywania minimum.
   $H \approx$ stałe na małych obszarach.

Uwaga

Ta analiza nie jest słuszna dla $F(\vec{x})$ zależnych liniowo od $\vec{x}$;
Wtedy: Klasa problemów – programowanie liniowe, rozwiązania leżą na krawędzi obszaru ograniczeń.

Kształt funkcji $F(x)$

Optymalne algorytmy minimalizacji

Nie ma uniwersalnej metody minimalizacji.
Dla bardzo złego algorytmu można znaleźć funkcję $F(x)$, którą minimalizuje on najszybciej – i odwrotnie.

Zasada doboru algorytmu optymalizacji

Zasada: Dla konkretnej funkcji należy indywidualnie dobrać algorytm minimalizacji.

Minimalizacja w 1-D

Minimalizacja w 1-D

Przydatność metod 1-D dla zag. n-D

- Prosta ilustracja ogólnych problemów
- Metody 1-D są często elementem składowym metod n-D

Przeglądanie siatki *(grid search)*

Realizacja

- Przegląd wszystkich elementów iloczynu kartezjańskiego podzbiorów parametrów.
- Wybór wartości najmniejszej.

Zalety

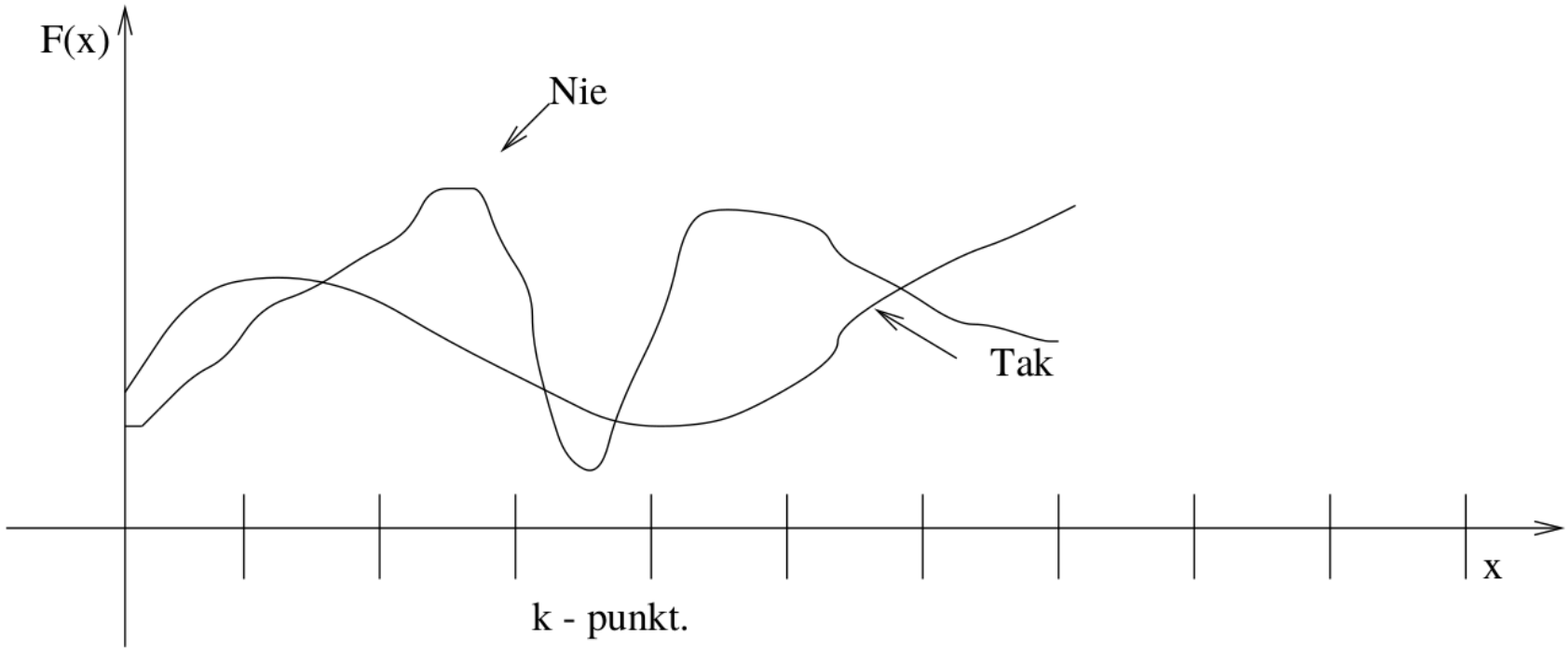
- Absolutna prostota, problem typu embarrassingly parallel.
- Bezwzględna zbieżność.
- Brak "czułości" na szczegółowe zachowanie się $F(x)$.

Wady

- Nie może być stosowana dla przedziału nieskończonego.
- Nieefektywna, nie "uczy się" własności funkcji.

Przeglądanie siatki *(grid search)*

Problem z doborem rozmiaru siatki, tak aby nie zgubić szukanej wartości.



Przeglądanie siatki *(grid search)*

Założenie: Poszukujemy minimum w k-D dysponując k zbiorami parametrów po 10 000 elementów każdy.

$$\begin{array}{lll} & 100 \text{ punktów} & \text{w } 1-\text{D} \\ \text{Zawężamy obszar do } 1\% \Rightarrow & 100^2 & \text{w } 2-\text{D} \\ & 100^{10} & \text{w } 10-\text{D} \end{array}$$

Przy czasie obliczeń jednej wartości $F(x) \approx 10^{-5} s$

$$\text{Czas obliczeń: } t_o = \frac{10^{20} * 10^{-5} s}{\underbrace{\pi * 10^7}_{\text{sek. w roku}}} \approx 3 * 10^7 lat!$$

Metoda złotego podziału *(golden section search)*

Definicja funkcji unimodalnej

Funkcję $f(x)$ nazywamy *unimodalną* na przedziale $[a,b]$ jeżeli:

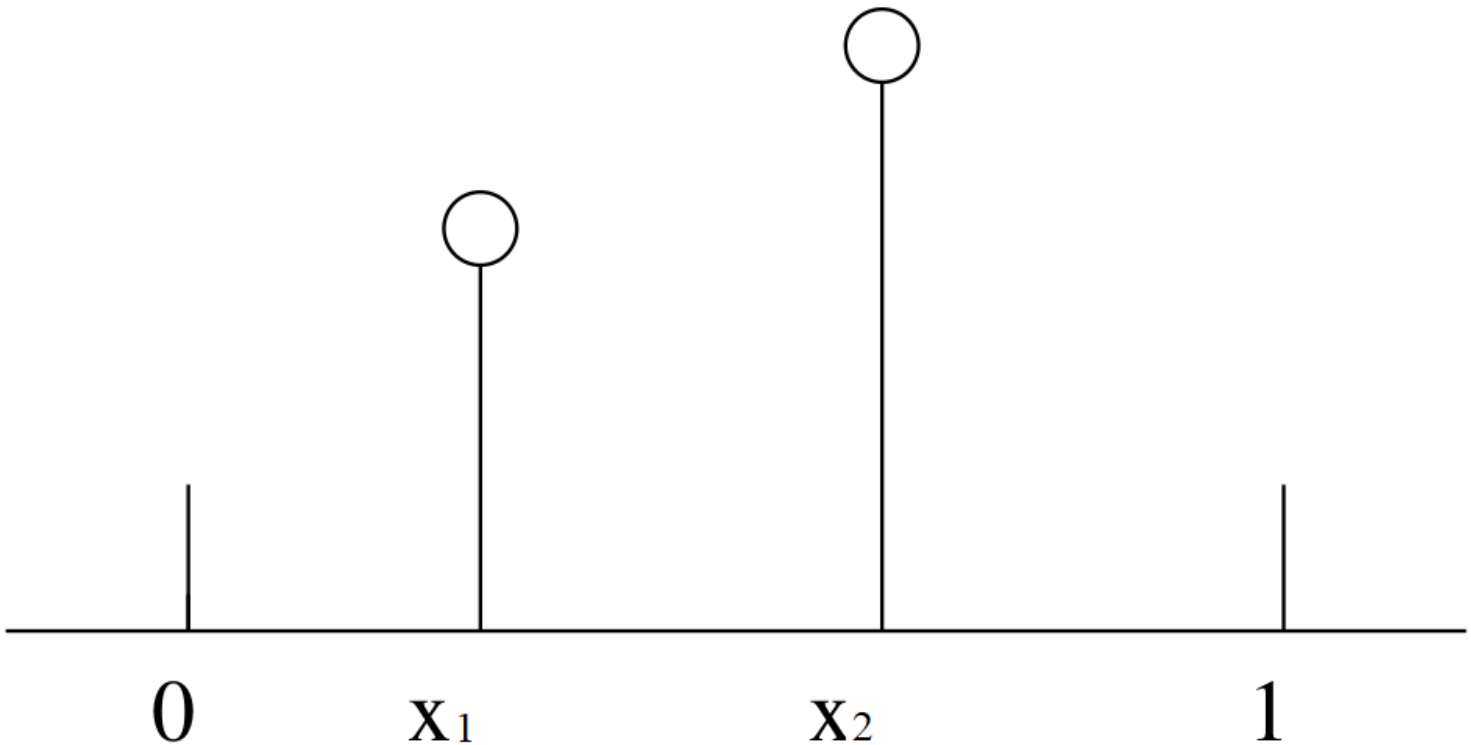
1. $\exists x^* \in [a,b] : f(x^*) = min_{x\in[a,b]} f(x)$
2. $\forall x_1,x_2 : a \leqslant x_1 < x_2 \leqslant b$ zachodzi:
   - $x_2 \leqslant x_* \Rightarrow f(x_1) > f(x_2)$
   - $x_1 \geqslant x_* \Rightarrow f(x_1) < f(x_2)$

czyli minimum w przedziale $[a, b]$ jest tyko jedno.

Założenia

Dane: f – funkcja unimodalna na przedziale $[a, b]$
Szukane: minimum x_0 funkcji f na przedziale $[a, b]$

Metoda złotego podziału *(golden section search)*



$F(x)$ - f. unimodalna

Metoda złotego podziału *(golden section search)*

Realizacja

$t \in (0,1)$ – współczynnik redukcji po każdym etapie

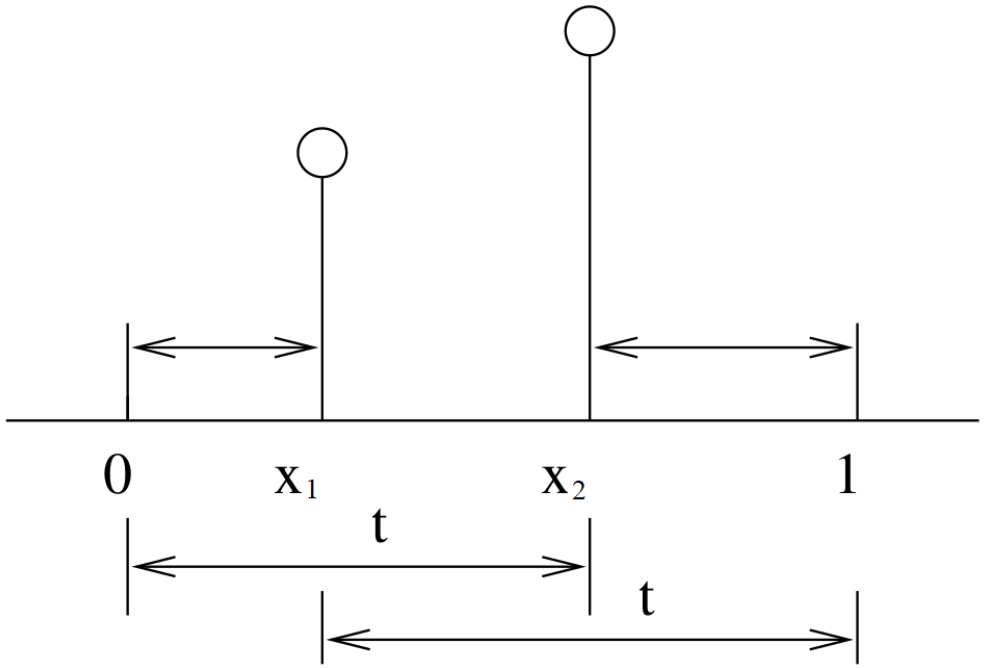
1. Obliczamy nową długość przedziałów $d = t(b-a)$
2. $x_L = b - d, x_R = a + d$
3. Jeżeli $f(x_L) > f(x_R) \Rightarrow x_0 \in [x_L, b]$, $a := x_L$,
   Jeżeli $f(x_L) < f(x_R) \Rightarrow x_0 \in [a, x_R]$, $b := x_R$,
4. Procedurę powtarzamy, aż do osiągnięcia żądanej zbieżności.

Minimalizacja ewaluacji

Problem: Chcemy zminimalizować liczbę ewaluacji funkcji f.
Rozwiązanie: Dobieramy współczynnik t, aby w kolejnym kroku wykorzystać jedną z dwóch próbek: $f(x_L)$ lub $f(x_R)$.

Metoda złotego podziału *(golden section search)*

Dla $[0, 1]$ – długość przedziału po pierwszym etapie: $d_1 = t$
Dla przykładowego rozmieszczenie punktów w kolejnym kroku rozwiązania szukamy w $(0, x_2)$:



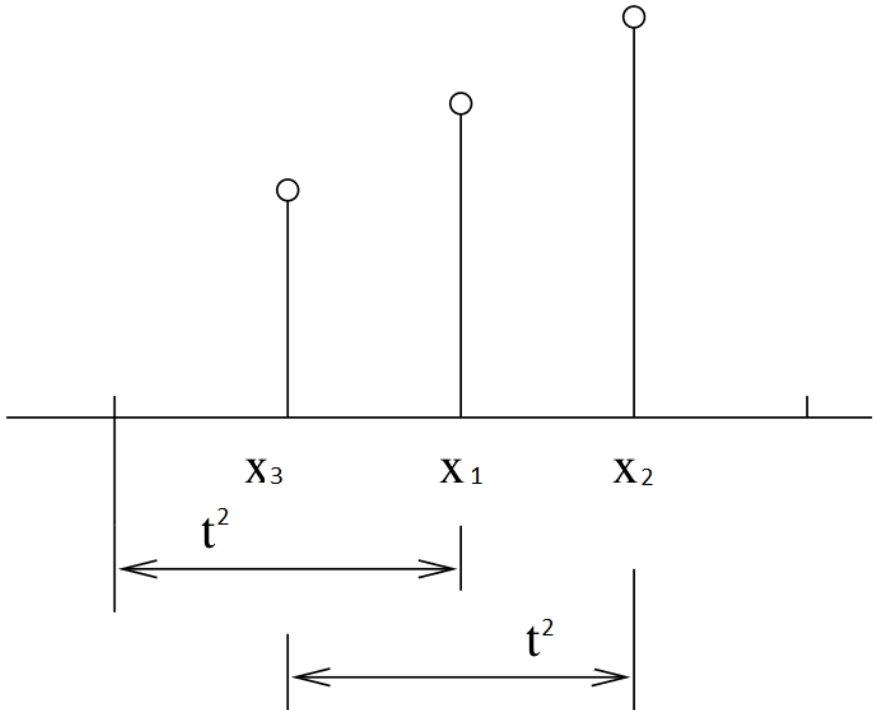
Chodzi o to, żeby w kolejnym kroku wykorzystać znaną już wartość, która zastała we wnętrzu przedziału (tutaj $f(x_1)$)

Metoda złotego podziału *(golden section search)*

Długość przedziału po drugim etapie to t^2. Jednocześnie długość ta jest wyznaczona przez pozycję z wnętrza przedziału na poprzednim etapie (tutaj pozycję x_1)

$$t^2 = 1 - t \Rightarrow t = \frac{\sqrt{5}-1}{2} \approx 0,616 \rightarrow \textit{złoty podział}$$

Dla zadanej liczby kroków – optymalna.



Kwadratowa interpolacja i aproksymacja

Założenia

Wykres funkcji $f(x)$ jest parabolą.

Realizacja

- Interpolujemy $f(x)$ funkcją kwadratową w 3 punktach: x_1, x_2, x_3.
- ekstremum $f(x)$ to ekstremum paraboli przechodzącej przez x_1, x_2, x_3, znajduje się w punkcie x_4:

$$x_4 = -\frac{\frac{f_1 \cdot (x_2+x_3)}{(x_1-x_2)\cdot(x_1-x_3)} + \frac{f_2 \cdot (x_1+x_3)}{(x_2-x_1)\cdot(x_2-x_3)} + \frac{f_3 \cdot (x_1+x_2)}{(x_3-x_1)\cdot(x_3-x_2)}}{2 \cdot \left[\frac{f_1}{(x_1-x_2)\cdot(x_1-x_3)} + \frac{f_2}{(x_2-x_1)\cdot(x_2-x_3)} + \frac{f_3}{(x_3-x_1)\cdot(x_3-x_2)}\right]}$$

Kwadratowa interpolacja i aproksymacja

Realizacja cd.

- x_4 – zastępuje jeden z x_1,x_2,x_3, wyznaczamy nowy x_4.
- Procedurę kończymy gdy wartość $f(x_4)$ jest bliska $f(x_3)$ z zadaną dokładnością.

Problemy

1. Na każdym kroku x_1,x_2,x_3 mogą wyznaczać *max*, a nie *min* powodując rozbieżność.
2. Gdy x_1,x_2,x_3 leżą prawie na prostej, otrzymujemy duży krok:
   - Trudności numeryczne
   - Rozbieżność
3. Który z poprzednich punktów odrzucić?
4. Możliwe oscylacje wokół minimum, zamiast zbieżności.

Kwadratowa interpolacja i aproksymacja

Możliwe zabezpieczenia

Zaniechanie metody przy wystąpieniu trudności.

Zastosowanie

W ostatniej fazie minimalizacji funkcji f.
(Funkcje fizyczne są zwykle paraboliczne w pobliżu minimum.)

Metoda prób i błędów *(success-failure method)*

Założenie

Procedura składa się z dwóch części:

- Iteracyjne zawężenie przedziału – podobne do grid search.
- Kwadratowa interpolacja na otrzymanym przedziale.

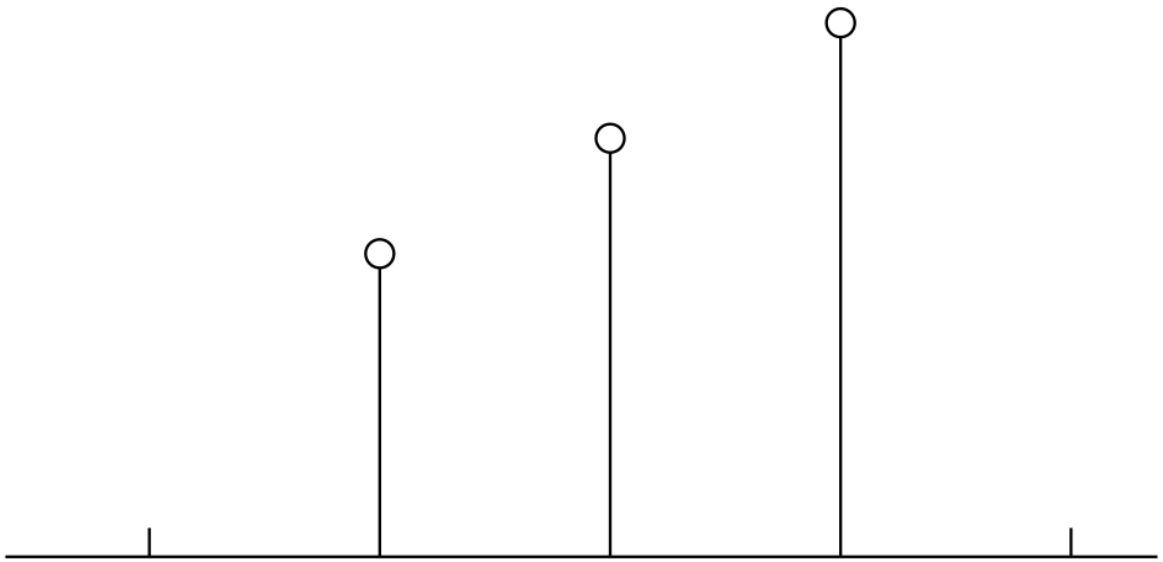
x_0 – start point, d – initial step size

- Gdy $f(x_0 + d) < f(x_0)$ – sukces:
  $x_0 \longrightarrow x_0 + d$,
  $d \longrightarrow \alpha * d, \alpha$ – expansion factor $(\alpha > 1)$
- Gdy $f(x_0 + d) > f(x_0)$ – niepowodzenie:
  $d \longrightarrow -\beta * d, \beta$ – contraction factor $(\beta < 1)$

α oraz β – ustalamy arbitralnie
Procedurę powtarzamy do zbieżności, tj. $|f(x_0 + d) - f(x_0)| < \epsilon$.

Metoda prób i błędów *(success-failure method)*

Minimum jest wstępnie zlokalizowane (bracketed), gdy po sukcesie - niepowodzenie, wtedy mamy trzy punkty typu:



Są one *punktem startowym* interpolacji kwadratowej.
Uniwersalna, efektywna metoda 1-D dla ogólnych funkcji.

Metody krokowe dla n-D (stepping methods in many variables)

Przeszukiwanie losowe

Problem: Grid search na n-D – złożoność czasowa $O(k^n)$.
Algorytm niepraktyczny dla $n > 2$.
Rozwiązanie: Metody Monte-Carlo – Wybór próbek losowo.

Realizacja

- $\vec{x_i}$ - ustalane losowo, zgodnie z rozkładem:
  - równomiernym,
  - normalnym
- Wybranie nalepszego znalezionego punktu.

Stosowane, gdy:

- nic nie wiadomo o $F(x)$.
- $F(x)$ ma kilka minimów.
- dla ustalenia rozsądnego punktu startowego innych metod.

Zmiana jednego parametru

Założenie

Funkcja f – ciągła na poszukiwanym obszarze.

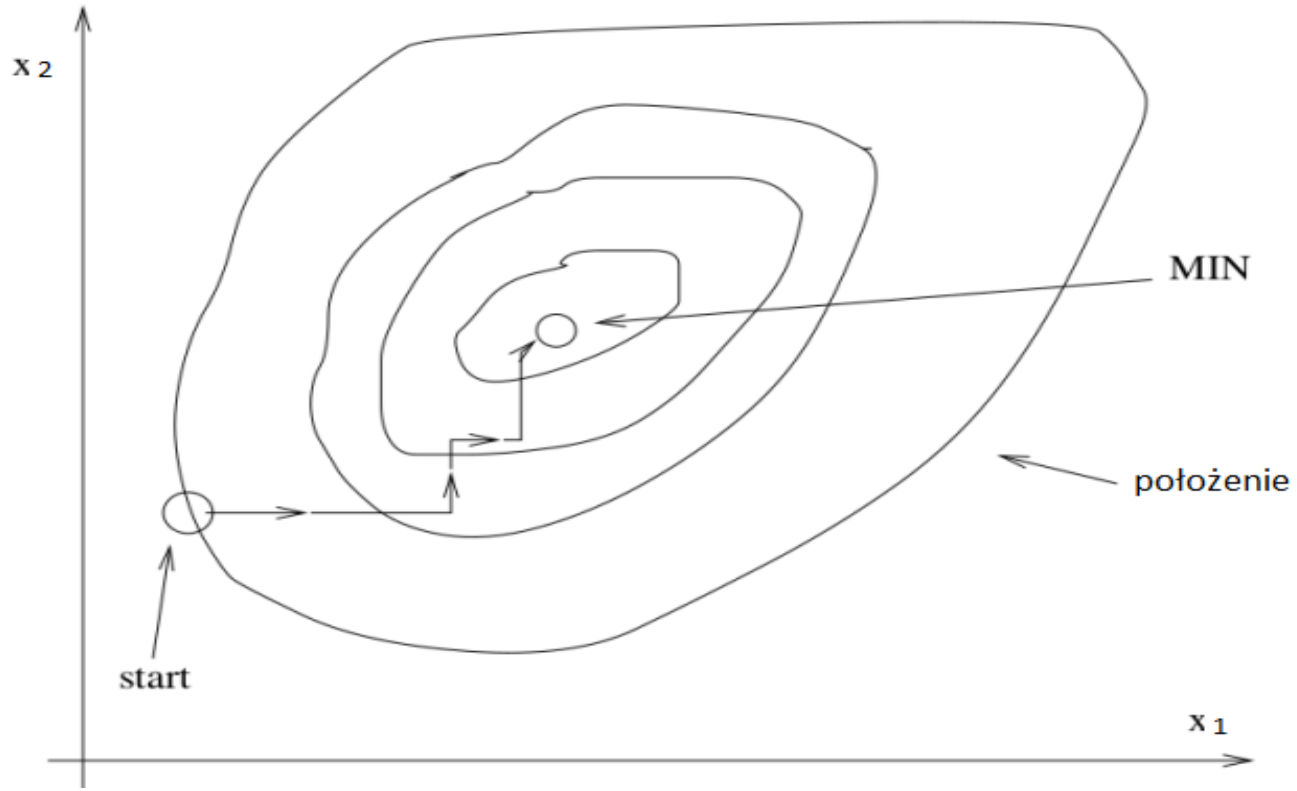
Obserwacja

Warunek na istnienie minimum - x_i – *stacjonarny punkt*, tj. znikają wszystkie n pochodne cząstkowe, $\frac{\partial f}{\partial x_i} = 0$, $i = 1, 2, 3, \ldots, n$

1. Wybierz jeden z n wymiarów.
2. Wykonaj optymalizację 1-D na wybranym wymiarze.
3. Powtarzaj do uzyskania punktu x_i, będącego minimum dla wszystkich wymiarów n.

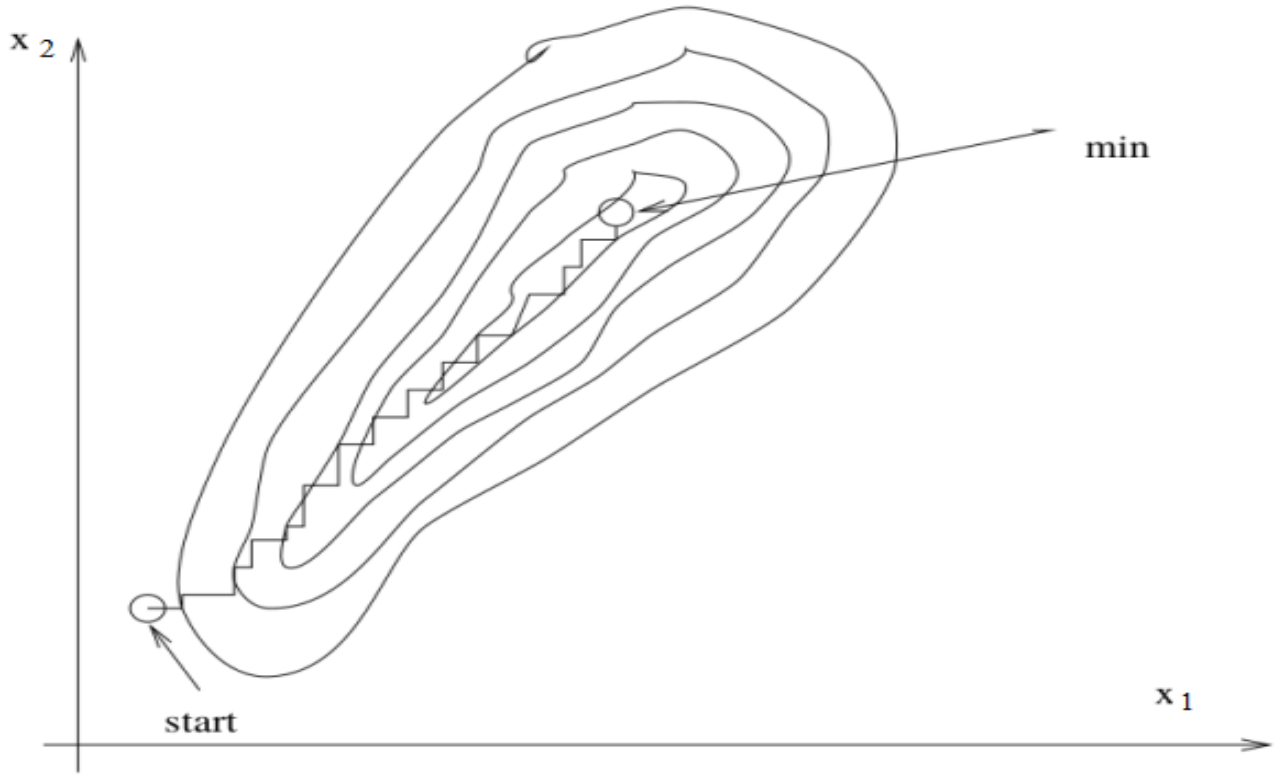
Zmiana jednego parametru cd.

Przykład dwuwymiarowy.



Zmiana jednego parametru cd.

Bardzo wolna z uwagi na przypadki z wąwozem (narrow valley):



Dla takiego przykładu stosujemy ulepszone metody.

Metoda Powella

Algorytm

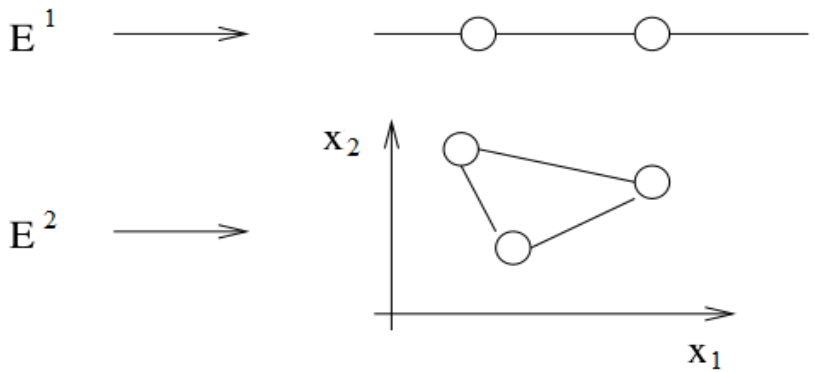
- Wykonaj jeden pełny cykl minimalizacji wzgl. kolejno wszystkich parametrów (współrzędnych),
- Zmień osie układu współrzędnych - nowy układ ortogonalny: jedna z osi od punktu początkowego do końcowego w ostatnim cyklu minimalizacji,
- Wykonaj kolejny cykl w nowym układzie współrzędnych.

Metoda mało efektywna dla dużego n

Simplex - wprowadzenie

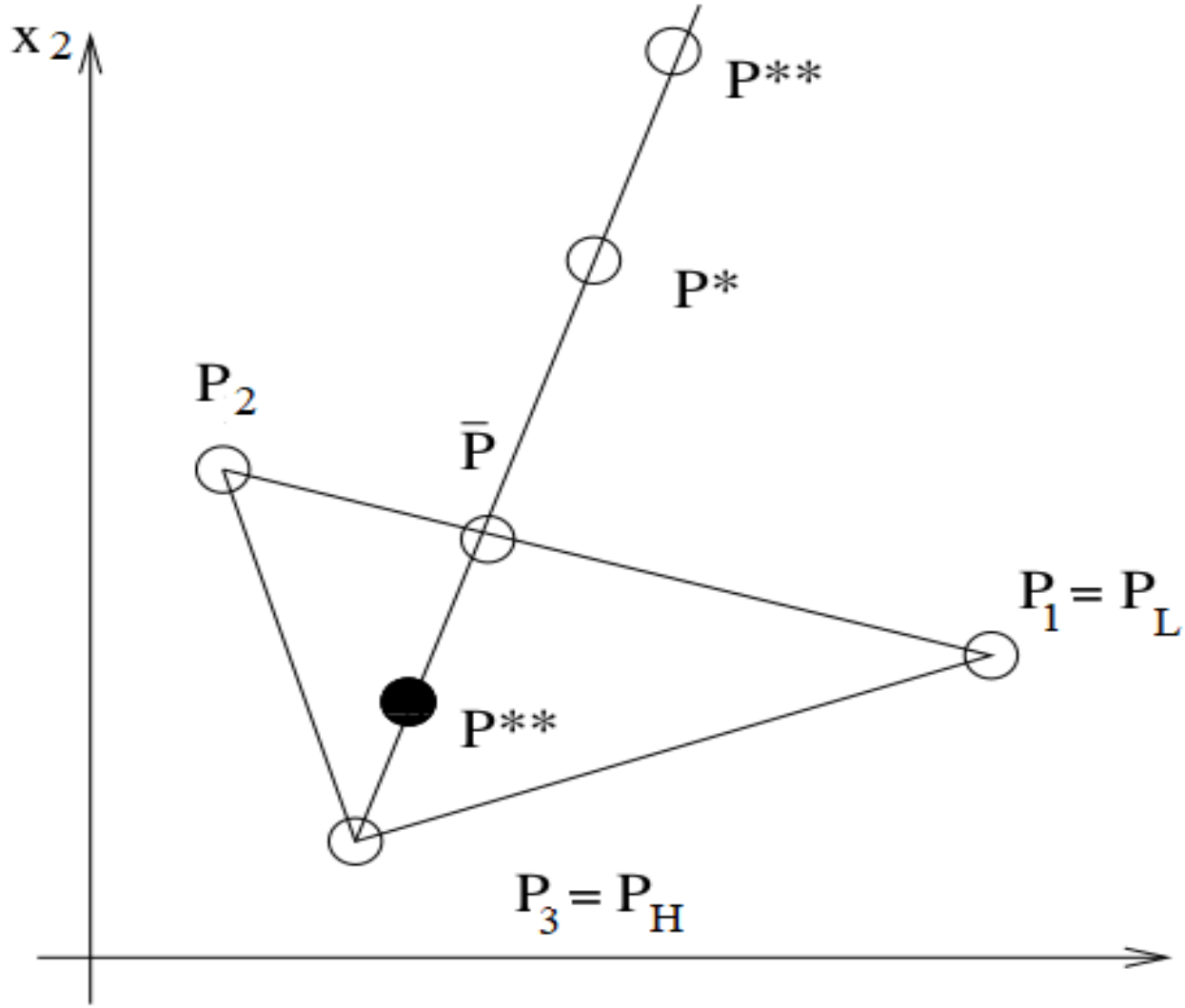
Definicja

Simplex - najprostsza n-wymiarowa figura określona przez $(n+1)$ wierzchołków (vertex).



Nazwa metody - w każdym kroku informacja o funkcji dotyczącej jej wartości w $n+1$ punktach

Opis metody cz.1



Opis metody cz.2

1. Wybieramy (losowo, min 1-D) 3 punkty P_1, P_2, P_3 i wyznaczamy spośród nich:
   P_H - gdzie F największa (highest)
   P_L - gdzie F najmniejsza (lowest)
2. Wyznaczamy "środek masy" wszystkich punktów z pominięciem P_H

$$\overline{P} = \frac{1}{n}\left[\sum_{i=1}^{n+1} P_i - P_H\right]$$

3. Obliczamy odbicie P_H wzgl. $\overline{P}$: $P^* = \overline{P} + (\overline{P} - P_H)$ jeżeli
   $F(P^*) < F(P_L) \Rightarrow$ nowy $P^{**} = \overline{P} + 2 * (\overline{P} - P_H)$
   $F(P^*) > F(P_L) \Rightarrow$ nowy $P^{**} = \overline{P} - 1/2 * (\overline{P} - P_H)$

Opis metody cz.3

4. Punkt P_H zastępujemy przez najlepszy z P^* i P^{**}

   Jeżeli żaden z nowych punktów nie jest lepszy od P_H, tworzymy simplex oparty o P_L w wymiarach $0.5 *$ poprzednie.

Modyfikacje:

- Inne współczynniki $\neq 2$ oraz $\neq \frac{1}{2}$,
- Interpolacja kwadratowa wzdłuż prostej $(P_H, \overline{P})$

Uwaga

Nowy punkt nie może być zbyt blisko $\overline{P}$, bo to grozi redukcją (bez powrotu) simpleksów w n do hiperpłaszczyzny.

Opis metody cz.4

Zalety :

- nieczuła na płytkie minima (pochodzenia: zaokrąglenia, statystyka . . .),
- mała ilość obliczeń funkcji $F(X)$ w każdym kroku,
- największe możliwe kroki,
- rozsądny kierunek poszukiwań,
- bezpieczna i szybka daleko od minimum

Zbieżność

$$\underbrace{EDM}_{\text{estimated distance to minimum}} = F(P_H) - F(P_L) < \epsilon$$

Metody gradientowe

Działają w oparciu o informacje o funkcji w małych obszarach (używany gradient i ew. wyższe pochodne).

Wyznaczanie pochodnych

Analitycznie – kłopotliwe, więc numerycznie:

$$\left.\frac{\partial(F)}{\partial(x)}\right|_{x_0} \approx \frac{F(x_0+d)-F(x_0)}{d}, \quad \delta \approx \frac{d^2}{2}\cdot\left.\frac{\partial^2 F}{\partial x^2}\right|_{x_0}$$

Lepiej:

$$\left.\frac{\partial(F)}{\partial(x)}\right|_{x_0} \approx \frac{F(x_0+d)-F(x_0-d)}{2\cdot d}, \quad \delta \approx \ldots\cdot\left.\frac{\partial^3 F}{\partial x^3}\right|_{x_0}$$

Wyznaczanie pochodnych cd.

- łatwo przy okazji obliczyć drugie pochodne:

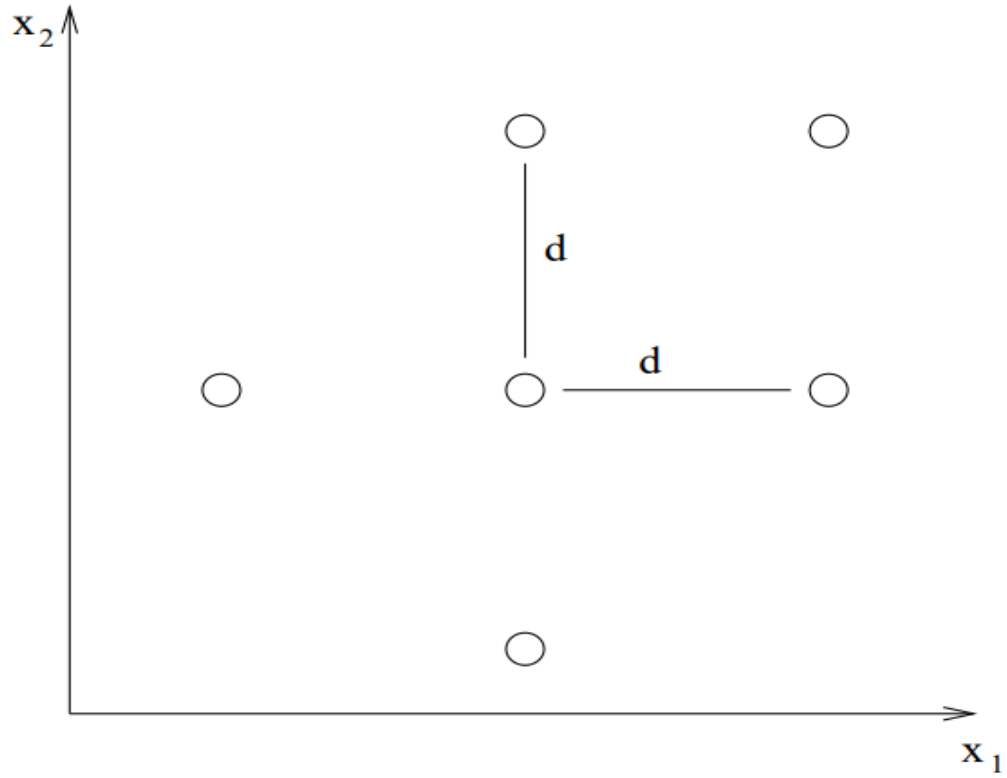
$$\left.\frac{\partial^2 F}{\partial x^2}\right|_{x_0} \approx \frac{F(x_0+d) - 2\cdot F(x_0) + F(x_0-d)}{d^2}$$

- Drugie pochodne tworzą macierz $n \times n$. $H_{ij} = \frac{\partial^2 F}{\partial x_i \partial x_j}$
  Na przykład:

$$\left.\frac{\partial^2 F}{\partial x \partial y}\right|_{x_0,y_0} \approx$$

$$\frac{F(x_0+d, y_0+d) - F(x_0, y_0+d) - F(x_0+d, y_0) + F(x_0, y_0)}{d^2}$$

Wyznaczanie pochodnych cd.

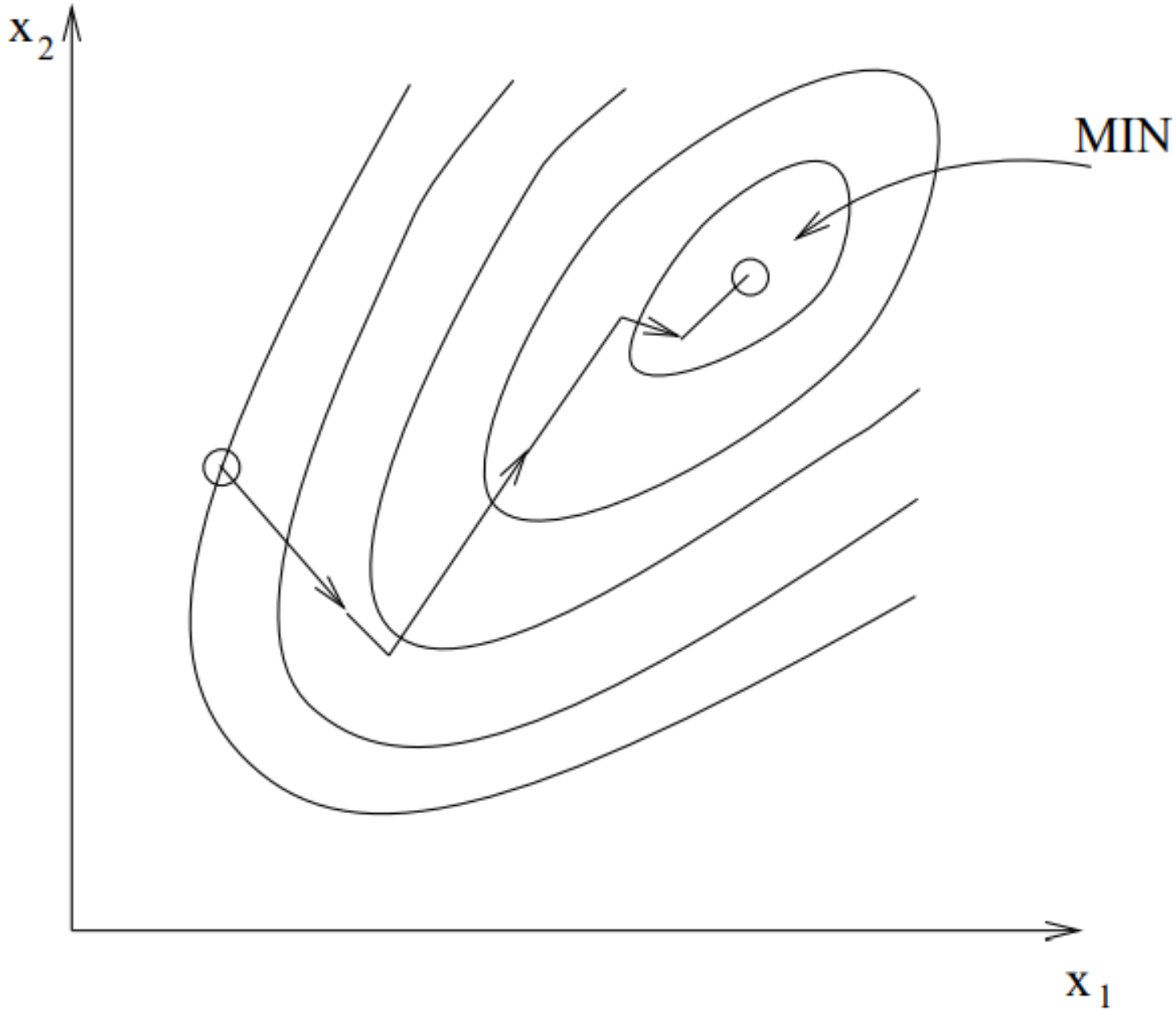


Ogólnie dla mieszanych pochodnych poza p. symetrycznymi potrzeba $\frac{n(n-1)}{2}$ pkt.

Metody gradientowe

Metoda największego spadku

- podążanie w kierunku wyznaczonym przez $-\vec{g}$ (gradient $g_i = \frac{\partial F}{\partial x_i}$)
- w tym kierunku funkcja maleje najszybciej (nierówność Cauchy'ego-Schwarza)
- seria minimalizacji 1-D wzdłuż kierunku największego spadku $\vec{x_{k+1}} = \vec{x_k} - \alpha_k \vec{g}|_{x_k}$; α_k dobrane tak, ze $f(\vec{x_k} - \alpha_k \vec{g}|_{x_k}) = min_{\alpha>0} f(\vec{x_k} - \alpha \vec{g}|_{x_k})$
- jeśli α - stałe $\rightarrow$ metoda gradientu prostego (vanilla)
- iteracyjna, bo gradient nie jest stały
- $g_{k+1} \perp g_k$ - metoda prowadzi do "zygzakowania"


x 2
MIN
x 1

Metoda Newtona - przypomnienie 1D

Znamy już metodę Newtona do znajdowania miejsc zerowych funkcji $f(x)$:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Metodę tę można zastosować do znajdowania możliwego minimum funkcji $f(x)$, którą przybliżamy wielomianem Taylora drugiego stopnia

$$f(y) = f(x) + f'(x)(y - x) + \frac{1}{2}f''(x)(y - x)^2$$

Następnie liczymy pochodną tego przybliżenia po y i szukamy jej miejsca zerowego metodą Newtona. Wzór iteracyjny przyjmuje postać:

$$x_{i+1} = x_i - \frac{f'(x_i)}{f''(x_i)}$$

Metody gradientowe

Metoda Newtona minimalizacji N-D

Ogólna funkcja kwadratowa jest określona przez:

- wartość,
- pierwsze pochodne
- drugie pochodne

} *W dowolnym punkcie* x_0 Dysponujemy tymi informacjami

$F(\vec{x})$ rozwijamy w szereg Taylora, pomijamy dalsze wyrazy:

$$F(\vec{x}) = F(\vec{x_0}) + \vec{g}^T \cdot (\vec{x} - \vec{x_0}) + \frac{1}{2}(\vec{x} - \vec{x_0})^T \cdot H \cdot (\vec{x} - \vec{x_0})$$

i szukamy minimum takiej funkcji kwadratowej
Wzór na minimum (iteracyjny, analogiczny do 1D):

$$\vec{x_{i+1}} = \vec{x_i} - H^{-1}|_{\vec{x_i}} \cdot \vec{g}|_{\vec{x_i}} = \vec{x_i} - V \cdot \vec{g}$$

Metoda Newtona cd.

Metoda jest n-D odpowiednikiem 1-D interpolacji kwadratowej.

Te same wady!

- może być niestabilna
- rozbieżna, gdy V nie jest dodatnio określona (czyli NIE zachodzi $\forall_{x \neq 0} x^T V x > 0$)

Zalety:

- krok nie jest dowolny, lecz określony przez metodę
- kierunek $\neq$ wartość gradientu, tylko brana pod uwagę korelacja parametrów (pamietane w macierzy V)

Metoda Newtona cd.

Używana:

- blisko minimum.
- gdy funkcja jest dodatnio określoną formą kwadratową.

Metoda Newtona jest podstawą wielu metod.

Dygresja: dodatnio określone formy kwadratowe

1-D forma kwadratowa:

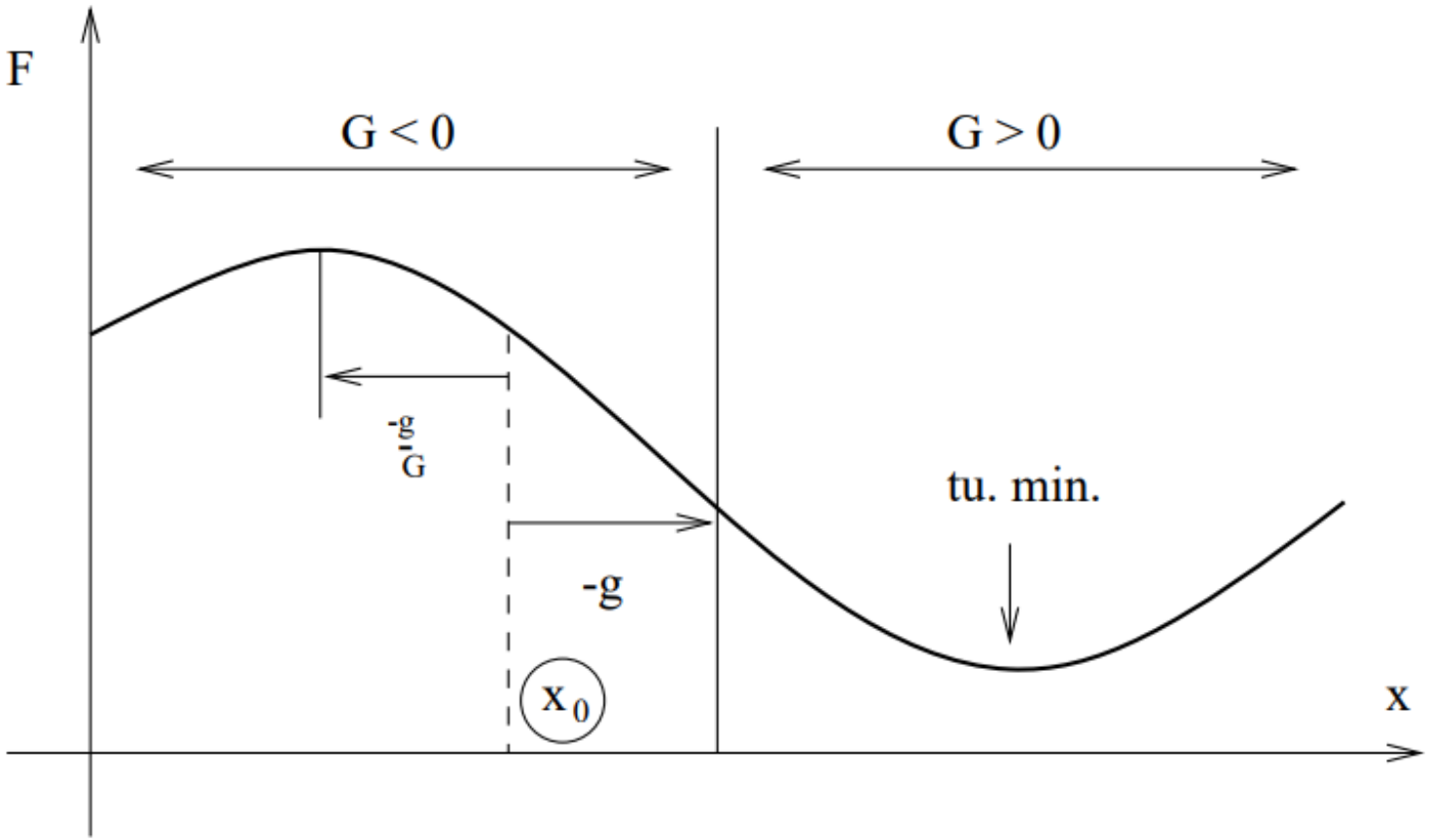
$$F(x) = a + g \cdot x + \frac{1}{2} G \cdot x^2$$

$$g = \left.\frac{\partial F}{\partial x}\right|_0, \qquad G = \left.\frac{\partial^2 F}{\partial x^2}\right|_0$$

$F(x)$ ma min. wtedy i tylko wtedy, gdy $G > 0$.

$$x_{min} = -g/G \qquad G = 0 \text{ to } x_{min} \to \infty;$$

Dodatnio określone formy kwadratowe cd.



Dodatnio określone formy kwadratowe cd.

Dla ogólnej funkcji nieliniowej:

- krok do $x = -\frac{g}{G}$ gdy $G > 0$
  (w przeciwnym przypadku: ∞ lub maximum)
- gdy $G \leqslant 0$ krok $= -g$
  $\longrightarrow$ kierunek - dobry,
  $\longrightarrow$ wartość - dowolna

W $x_0 \longrightarrow F(x)$ nie jest wypukła (dodatnio określona)

Uogólnienie na n-D:
$g \longrightarrow \vec{g}$, G - macierz 2-ich poch (inaczej hesjan H).;
$F(\vec{X}) = a + \vec{g}^T \cdot \vec{x} + \frac{1}{2}\vec{x}^T G$ tylko dla G dod. określonej ma sens
krok do: $\vec{x} = -G^{-1} \cdot \vec{g}$

$$\vec{x_{i+1}} = \vec{x_i} - G^{-1} \cdot \vec{g}$$

Metody gradientowe

Badanie czy G jest dodatnio określona.
Brak prostych sposobów.

Dla macierzy kwadratowych, symetrycznych - 2 warunki konieczne

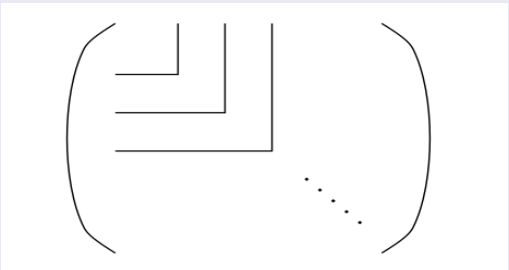
1^o elementy diagonalne > 0
2^o elementy pozadiagonalne:

$$G_{ij}^2 < G_{ii} \cdot G_{jj}$$

Badanie czy G jest dodatnio określona cd.

Ogólne warunki konieczne i wystarczające

- Definicja: skalar $\vec{e}^T \cdot G \cdot \vec{e} > 0$ dla każdego $\vec{e} \neq 0$ wyjaśnia dlaczego G dod. okr. daje formę $F(x)$ z minimum: $F(x)$ rośnie we wszystkich kierunkach od $\vec{e} = 0$
- wszystkie wartości własne > 0 (b. trudne i przybliżone)
- wyznacznik wszystkich macierzy > 0



(najprościej sprawdzić)

- $V = G^{-1}$ jest dodatnio określona

Metody gradientowe

Postępowanie w przypadku G - nie jest dod. określone

a) analogicznie do 1-D : G = I $\rightarrow$ Ale:

b) lepiej:

- gdy wszystkie elementy diagonalne $G > 0$, wtedy pozadiag. $\rightarrow = 0$ (scale - invariant step),
- gdy tylko niektóre el. pozadiag. $G_{ij}^2 \geqslant G_{ii} G_{jj} \Rightarrow G_{ij} = 0$,
- zamiast G^{-1} bierzemy $(G + \lambda \cdot I)^{-1}$;
  $\lambda \geqslant$ największa (bezwzgl.) ujemna wartość własna (dużo obliczeń)

Gdy G - nie jest dodatnio określone cd.

Metody oparte na powyższych regułach - *quasi-Newton method*

Podstawowa wada tych metod - obliczanie i odwracanie w każdym kroku macierzy drugich pochodnych

- obliczanie 2-ich poch. $\sim n^2$ (długie)
- odwracanie

Metody gradientowe

kierunki sprzężone *(conjugate directions)*

Wektory $\vec{d}_i$ i $\vec{d}_j$ *są sprzężone* ze względu na dodatnio określoną macierz, jeżeli:
$\vec{d}_i^{\,T} A \vec{d}_j = 0$ dla $i \neq j$;
gdy $A = I, \vec{d}_i \rightarrow$ ortogonalne
(sprzężenie - uogólnienie ortogonalności)
n sprzężonych wektorów rozpina n-D przestrzeń.
A - nie określa jednoznacznie zbioru wektorów sprzężonych

W minimalizacji użyteczne są wektory sprzężone ze względu na *hesjan* H.

Metoda sprzężonych kierunków cd.

Można pokazać, że

- Sekwencja liniowych minimalizacji w każdym z n sprzężonych kierunków minimalizuje ogólną funkcję kwadratową n zmiennych.
- Minimalizacja wzdłuż jednego z kierunków sprzężonych jest niezależna od minimalizacji względem pozostałych sprzężonych kierunków

Wnioski:

Minimalizacja nie wzdłuż osi ortogonalnych, ale wzdłuż kierunków sprzężonych

Metoda gradientów sprzężonych (conjugate gradients)

Jak wyznaczyć kierunki sprzężone bez znajomości H?

Metoda wykorzystuje tylko 1-sze pochodne.

Jeśli $F(\vec{x})$ i $\vec{g}(\vec{x})$ wyznaczone w $\vec{x_0}$ i $\vec{x_1}$, z nich:
$\vec{\Delta x} = \vec{x_1} - \vec{x_0}, \quad \vec{\Delta g} = \vec{g_1} - \vec{g_0}$
to dla $F(\vec{x})$ kwadratowej, z hesjanem H: $\boxed{\vec{\Delta g} = H \cdot \vec{\Delta x}}$,
zatem dowolny $\vec{d_1}$ sprzężony do $\vec{\Delta x}$ będzie $\perp \vec{\Delta g}$:

$$\boxed{\vec{d_1}^T H \vec{\Delta x} = \vec{d_1}^T \cdot \vec{\Delta g} = 0 (*)}$$

Pierwszy kierunek: $\vec{d_0} = -\vec{g_0}$
znajdujemy minimum $\vec{x_1}$ wzdłuż $\vec{d_0}$ według $\vec{x_1} = \vec{x_0} + \alpha \vec{d_0}$
a następnie gradient w tym punkcie (czyli $\vec{g_1}$)
Drugi kierunek tworzymy jako liniową kombinację znanych kierunków: $\vec{d_1} = -\vec{g_1} + b \cdot \vec{d_0}$

Metoda gradientów sprzężonych cd.

Jak ustalić b?

Warunek sprzężenia: $\vec{d_1}^T \cdot H \cdot \vec{d_0} = 0$, czyli $\vec{d_1}^T \cdot H \cdot \frac{1}{\alpha}(\vec{x_1} - \vec{x_0}) = 0$

z (*) $\vec{d_1}^T \cdot (\vec{g_1} - \vec{g_0}) = 0$ czyli: $(-\vec{g_1} - b \cdot \vec{g_0})^T \cdot (\vec{g_1} - \vec{g_0}) = 0$

$\vec{x_1}$ - jest znalezionym minimum wzdłuż $-\vec{g_0}$

$\Rightarrow$ kolejny $\vec{g_1}$, uzyskany w p.$\vec{x_1}$, więc jest $\perp$ do $\vec{g_0} \Rightarrow \vec{g_1}^T \cdot \vec{g_0} = 0$

$\Rightarrow \boxed{b = \frac{\vec{g_1}^T \cdot \vec{g_1}}{\vec{g_0}^T \cdot \vec{g_0}}}$, czyli nowy sprzężony kierunek

$\vec{d_1} = -\vec{g_1} + \left(\frac{\vec{g_1}^T \cdot \vec{g_1}}{\vec{g_0}^T \cdot \vec{g_0}}\right) \cdot \vec{d_0}$

Ten proces kontynuujemy generując n kierunków wzajemnie sprzężonych do użycia dla kolejnych kroków minimalizacji

$$\boxed{\vec{d_{i+1}} = -\vec{g_{i+1}} + \frac{\vec{g_{i+1}}^T \cdot \vec{g_{i+1}}}{\vec{g_i}^T \cdot \vec{g_i}} \cdot \vec{d_i}}$$

Literatura

- Uczenie maszynowe `https://ruder.io/optimizing-gradient-descent/index.html`
- Minimalizacja za pomocą kompiterów kwantowych `https://pennylane.ai/qml/demos/tutorial_quantum_natural_gradient.html`
- Scipy notes (dobry przegląd referencji) `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html`
- Scipy lectures `https://scipy-lectures.org/advanced/mathematical_optimization`