1D OPTIMIZACIJA

METODE DIREKTNOG PRETRAŽIVANJA

FIBONAČI (a, b, tol)

- Koristimo odnos fibonačijevih brojeva da podelimo zadati interval
- $\underline{\text{K.Z.}}$: znamo broj iteracija unapred $F_n > \frac{b-a}{\epsilon}$

(+)
- jednostavna implementacija
- koristan za veći interval i veći skup tačaka

(−)
- fja mora biti unimodalna na zadatom intervalu
- moramo unapred znati ponašanje fje kako bismo zadali interval

ZLATNI PRESEK (a, b, tol)

- Zlatni presek = $\varphi = \frac{3-\sqrt{5}}{2}$
- Funkcioniše isto kao zlatni presek, samo ne znamo broj iteracija u početku

GRADIJENTNE METODE

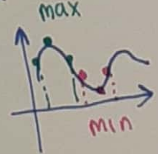
NJUTN-RAPSON (x0, tol)

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

$\underline{\text{K.Z.}}$: $|x_{k+1} - x_k| < \epsilon$

(+)
- Relativno brza (jedino je sečica brža)
- Jednostavna za implementaciju (može se računati analitički)

(−)
- Moramo znati I i II izvod. Fja mora biti bar 2 puta diferencijabilna.
- Određuje se samo ekstrem, moramo unapred znati ponašanje fje da bismo znali da li je min ili max
- Moramo dobro odabrati početnu tačku:



- Moguća divergencija

SEČICA $(x0, x1, tol)$

- Cilj: izbegavanje izvoda

$$\boxed{x_{k+1} = x_k + f'(x_k)\,\frac{x_k - x_{k-1}}{f'(x_k) - f'(x_{k-1})}}$$

K.Z.: $\boxed{|x_{k+1} - x_k| < \varepsilon}$

(+)
- brzina
- potrebno poznavanje samo I izvoda

(−)
- potrebno je pamtiti 2 tačke, a ne samo 1
- moguća divergencija

METOD APROKSIMACIJE POLINOMOM

- $f(x)$ aproksimiramo polinomom $g(x)$ i tražimo optimum polinoma $\boxed{g'(x) = 0}$

K.Z. $\boxed{|g(x_{opt}) - f(x_{opt})| < \varepsilon}$

CS CamScanner

VIŠEDIMENZIONA OPTIMIZACIJA

1. METOD NAJBRŽEG PADA

Gradijent = izvod po svim promenljivim
= normala na tangentu
= smer najbržeg porasta

$$\boxed{x_{k+1} = x_k - \gamma \nabla f(x_k)}$$

γ = korak

K.Z. : - $\boxed{\| \nabla f(x_k) \| < \varepsilon}$
- max broj iteracija

⊖ - velik broj iteracija
- preosetljiv na oblik fje
- može dostići toliki broj iteracija da dođe do nestabilnosti
- ako nađe dovoljno mali gradijent, tu se zaustavlja, ali možda je stigao do lokalnog ekstrema
- mali gradijent → mali korak, velik gradijent - velik korak, a poželjno je obrnuto

2. METOD NAJBRŽEG PADA SA MOMENTOM

$$\boxed{\begin{aligned} v_k &= \omega v_{k-1} + \gamma \nabla f(x_k) \\ x_{k+1} &= x_k - v_k \end{aligned}}$$

ω = inercija

⊕ - Neće se zaustaviti nego ako je fja bila strma, nastaviće da se kreće zbog akumulacije inercije
- Smanjuje oscilacije

⊖ - I dalje spora, ali smer je više kontrolisan
- Moramo da vodimo računa o 2 promenljive : ω i γ

CamScanner

3. UBRZANI GRADIJENT NESTEROVA

$$\boxed{\begin{aligned} x'_k &= x_{k-1} - \omega v_{k-1} \\ v_k &= \omega v_{k-1} + \gamma \nabla f(x'_k) \\ x_{k+1} &= x_k - v_k \end{aligned}}$$

- Koristimo buduću tačku

(+) - direktniji, manje osciluje

(−) - moramo da vodimo računa o parametrima

4.

CamScanner

Njutn - Rapson

- Tražimo ekstrem tj. stacionarnu tačku, ne znamo da li je min ili max

$$\boxed{f'(x) = 0}$$

- razvoj u Tejlorov red:

$$f(x) = \underbrace{f(x_0)}_{a} + \underbrace{f'(x_0)(x-x_0)}_{b(x-x_0)} + \underbrace{\frac{1}{2} f''(x_0)(x-x_0)^2}_{\frac{1}{2}c(x-x_0)^2} = g(x)$$

$$g'(x) = f'(x_0) + f''(x-x_0) = 0$$

$$f''(x_0)(x-x_0) = -f'(x_0)$$

$$x - x_0 = -\frac{f'(x_0)}{f''(x_0)}$$

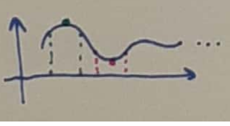
$$x = x_0 - \frac{f'(x_0)}{f''(x_0)}$$

$$\boxed{x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}}$$

- Ovaj algoritam se ponavlja dok se ne zadovolji tolerancija

$$\boxed{|x_{k+1} - x_k| \leq tol}$$

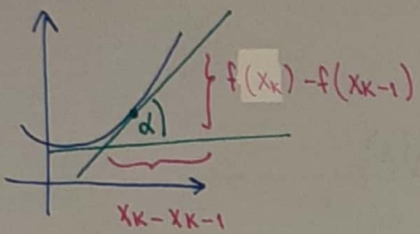
- MANA: potreban nam je I i II izvod. Da bih mogla da primenim ovaj metod fja mora da bude diferencijabilna bar do 2. reda i izvodi nam moraju biti poznati
- MANA: mora nam unapred biti poznata priroda problema tj. kako funkcija izgleda jer ova metoda traži ekstrem, ne znamo da li je min ili max
- MANA: moramo dobro da odaberemo početnu tačku

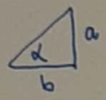


CamScanner

Sečica

- Izbegavamo drugi izvod koji nam je neophodan kod Njutn-Rapsona
- Izvod predstavlja k (nagib) tangente





$$tg\,\alpha = \frac{a}{b}$$

$$tg\,\alpha = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} = f'(x_k)$$

$$f''(x_k) = \frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}}$$

$$\boxed{x_{k+1} = x_k - f'(x_k)\,\frac{x_k - x_{k-1}}{f'(x_k) - f'(x_{k-1})}}$$

<u>MANA</u>: Izbegli smo drugi izvod, ali sada umesto čuvanja jedne prethodne vrednosti, moramo da čuvamo dve

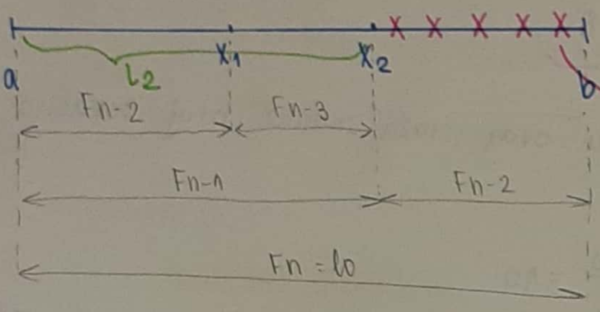
- kriterijum zaustavljanja: $\boxed{|x_{k+1} - x_k| < tol}$

CamScanner

FIBONAČIJEV METOD

$F_n = F_{n-1} + F_{n-2}$ svaki broj dobijamo kao zbir prethodna dva

1, 1, 2, 3, 5, 8, 13 ...



l_0 = dužina intervala

$l_0 = a+b$

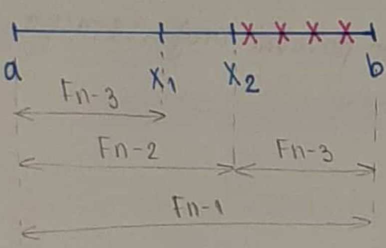
pp : tražimo min

ε : zadata tačnost

izbacujemo interval $L_2^* = \frac{F_{n-2}}{F_n} l_0$

- Fja je na tom intervalu rastuća pa tu sigurno nije min

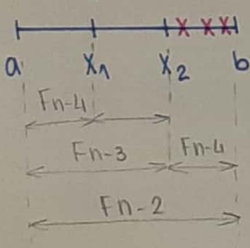
uzimamo interval $L_2 = \frac{F_{n-1}}{F_n} l_0$



izbacujemo: $L_3^* = \frac{F_{n-3}}{F_{n-1}} L_2$

uzimamo: $L_3 = \frac{F_{n-2}}{F_{n-1}} L_2 = \frac{F_{n-2}}{\cancel{F_{n-1}}} \cdot \frac{\cancel{F_{n-1}}}{F_n} l_0$

$L_3 = \frac{F_{n-2}}{F_n} l_0$



odbacujemo:

$L_4^* = \frac{F_{n-4}}{F_{n-2}} L_3$

uzimamo:

$L_4 = \frac{F_{n-3}}{F_{n-2}} L_3$

$L_4 = \frac{F_{n-3}}{F_n} l_0$

$x_1 = a + \frac{F_{n-2}}{F_n} l_0$

$x_2 = a + \frac{F_{n-1}}{F_n} l_0 = b - \frac{F_{n-2}}{F_n} l_0$

$x_1 + x_2 = a + \cancel{\frac{F_{n-2}}{F_n} l_0} + b - \cancel{\frac{F_{n-2}}{F_n} l_0}$

$x_1 + x_2 = a + b$

$\boxed{x_2 = a + b - x_1}$

$x_1 = a + \frac{F_{n-3}}{F_{n-1}} L_2 = a + \frac{F_{n-3}}{\cancel{F_{n-1}}} \cdot \frac{\cancel{F_{n-1}}}{F_n} l_0$

$x_1 = a + \frac{F_{n-3}}{F_n} l_0$

$x_2 = a + b - x_1$

$x_1 = a + \frac{F_{n-4}}{F_{n-2}} L_3 = a + \frac{F_{n-4}}{F_n} l_0$

$x_2 = a + b - x_1$

Fibonači u opštim brojevima:

$$L_n = \frac{\overset{=1}{F_1}}{F_n} \cdot L_0 = \frac{1}{F_n} L_0$$

L_n = dužina intervala posle n iteracija

L_0 = početna dužina

- Prvi fibonačijev broj $F_1 = 1$

Kriterijum zaustavljanja:

$$L_n < \varepsilon \Rightarrow \varepsilon > \frac{L_0}{F_n} \Rightarrow \underbrace{F_n > \frac{L_0}{\varepsilon}}$$

- Na osnovu ovog izraza znamo broj iteracija unapred

npr: $\frac{L_0}{\varepsilon} = 10$

Prvi fib. broj >10 je 13 tj. 7. po redu

$\Rightarrow n = 7$

Zlatni presek

MANA: ne znamo početni broj iteracija

$$\frac{F_n}{F_{n-1}} = \varphi \rightarrow \text{Fibonačijev broj}$$

$$\frac{F_{n-2}}{F_n} = \underbrace{\frac{F_{n-1}}{F_n}}_{\frac{1}{\varphi}} \cdot \underbrace{\frac{F_{n-2}}{F_{n-1}}}_{\frac{1}{\varphi}} = \frac{1}{\varphi^2}$$

$$\frac{F_{n-3}}{F_n} = \frac{1}{\varphi^3}$$

$$\boxed{\frac{F_{n-k}}{F_n} = \frac{1}{\varphi^k}}$$

Metod najbržeg pada

$f(x) = (x_1, x_2 \ldots x_n)$

- Gradijent predstavlja pravac najbržeg uspona, negativni gradijent predstavlja pravac najbržeg pada. Njega koristimo da pronađemo minimum višedim fje. U svakoj iteraciji tekuće rešenje se pomera u pravcu neg. grad.

$$\nabla f = \left[\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \quad \ldots \quad \frac{\partial f}{\partial x_n} \right]^T$$

- Algoritam:

1. Inicijalizacija:

   x_0 = početno pogađanje

   $\gamma > 0$ = veličina koraka

   $\varepsilon > 0$ = tolerancija

   N = maksimalan br koraka

2. Iteracije:

   $$x_{k+1} = x_k - \gamma \nabla f(x_k)$$

3. Kriterijum zaustavljanja:

   $\|\nabla f(x_k)\| \leq \varepsilon$ ili dostižemo max br iteracija (N)

- Lokalni optimumi su problem za sve grad. metode

Gradijent sa momentom

$$v_k = \omega v_{k-1} + \gamma \nabla f(x_k)$$

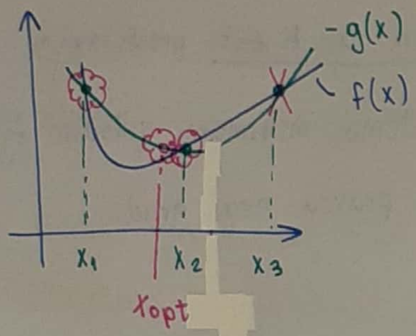
$$x_{k+1} = x_k - v_k$$

INERCIJA → PAMTIMO PRETHODNI KORAK

- Smanjuje oscilacije
- Sposoban da prođe kroz prev. tačke za razliku od najbržeg pada

CamScanner

Aproksimacija polinomom



$f(x) \approx ax^2 + bx + c$

Tražimo optimum parabole kojom smo aproksimirali funkciju $f(x)$

$$\left.\begin{aligned} a x_1^2 + bx_1 + c &= f(x_1) \\ a x_2^2 + bx_2 + c &= f(x_2) \\ a x_3^2 + bx_3 + c &= f(x_3) \end{aligned}\right\} \rightarrow a,b,c \rightarrow g(x) = ax^2 + bx + c$$

$g'(x) = 0 \leftarrow$ optimum

$g'(x) = 2ax + b = 0$

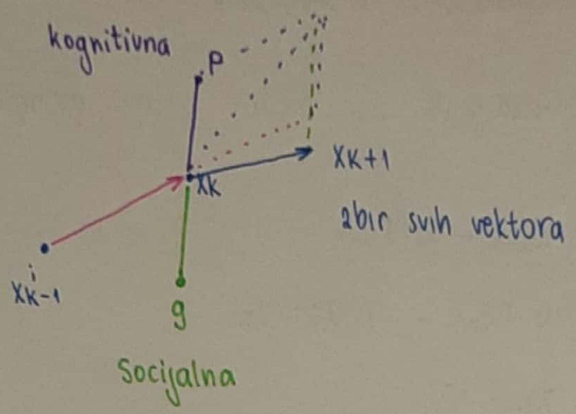
$x = -\frac{b}{2a} = x_{opt}$

- Ponavljamo postupak samo što menjamo tačke svaki put.

Kriterijum zaustavljanja : $\boxed{|g(x_{opt}) - f(x_{opt})| < \varepsilon}$

CamScanner

Optimizacija rojem čestica (PSO)



- **čestica** = potencijalno rešenje
- **roj** = skup čestica
- **Pbest** = personal best
- **gbest** = global best
- Čestice imaju sposobnost memorisanja pbest, komunikacije sa drugim česticama i gbest se pamti na nivou celog roja

Algoritam

1. Razbacamo čestice na random početne pozicije. Na taj način svaka čestica će pretražiti po jedan deo i naći globalni minimum, a neće stati kod nekog lokalnog koji nije najbolje rešenje.

2. $x_{k+1}^i = x_k^i + v_{k+1}^i$

k = br iteracija
i = indeks određene čestice

$$v_{k+1} = \underbrace{w \cdot v_k^i}_{\text{inercija}} + \underbrace{c_p r_p (p^i - x_k^i)}_{\text{kognitivna komponenta}} + \underbrace{c_g r_g (g - x_k^i)}_{\text{socijalna komponenta}}$$

inercija - na kretanje utiče to kako se čestica ranije kretala

kognitivna komponenta - "pojedinačno" najbolje iskustvo

socijalna komponenta - globalno najbolje rešenje

w = faktor inercije $0.9 \to 0.4$
c_p = kognitivni faktor $2.5 \to 0.5$
c_g = socijalni faktor $0.5 \to 2.5$
$r_p, r_g \in$ rand $[0,1]$

3. Ponavljamo 2. unapred određen broj iteracija

GENETSKI ALGORITAM

jedinka = potencijalno rešenje

f-ja prilagođenosti = k.o.

populacija = skup jedinki

generacija = iteracija

Jedinke se sastoje od gena = hromozoma

- GA opisuje evoluciju tj. prirodnu selekc.

ALGORITAM

1. Inicijalizacija → N jedinki
   - Razbacane jedinke na prostoru koji treba da pretražimo (domenu)
   - Na taj način pretražujemo ceo prostor, a ne samo neke delove
2. Selekcija → $\frac{N}{2}$ parova roditelja (*)
   - sortiranje, tiket, rulet, rulet sa rangiranjem ...
3. Ukrštanje → N dece (potomaka) (*)
   - svaki par roditelja ima par dece ukupno: 2N

4* Mutacija → "mutirane" jedinke (*)
   - ako je ispunjen uslov

5. Elitizam → N jedinki
   - Biramo N jedinki da "žive", N uklanjamo. Preživeće one jedinke koje su bolje prilagođene

PONAVLJAMO 2-5 dok se ne ispuni kriterijum

6. Nova generacija → najbolja jedinka
   - optimum

(*) Ruletska selekcija sa rangiranjem

x	f(x)	rank	rand	score
A	5	2	0.8	1.6
B	2500	4	0.2	0.8
C	500	3	0.1	0.3
D	0.01	1	0.4	0.4

- Roditelji su oni koji imaju najbolji score.
- Svaka jedinka ima šansu da bude roditelj, makar ona bila mala, nije = 0
- Na ovaj način izbegavamo pojavu superjedinki

(*) Binarno ukrštanje

r

r_1 : 0010|1101

r_2 : 0101|1010

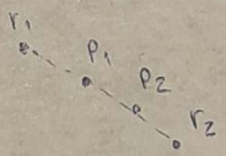
p_1 : 0010 1010

p_2 : 0101 1101

ukrštanje u jednoj tački:

r = rand int (0, dužina jedinke)

(*) Realno ukrštanje



$a = rand(0,1)$

$p_1 = a r_1 + (1-a) r_2$

$p_2 = (1-a) r_1 + a r_2$

(*) Binarna mut.

1) if rand (0,1) < mut

0010 → 0110

2) inverzija

10111
↓
11101

(*) Realna mut.

Dmut = rand(-1,1)·a

a = širina mut.

③ Убрзани градијент Нестерова

⇒ користимо „будућу" тачку да би брже итерирали кроз алгоритам

$$x'_k = x_{k-1} - w v_{k-1}$$

$$v_k = w v_{k-1} + \gamma \nabla f(x'_k)$$

→ градијент будуће тачке

→ утицаће на тренутну

$$x_{k+1} = x_k - v_k$$

⇒ директније, мање осцилација

(−)
↳ морамо пазити на параметре

④ ADAGRAD

⇒ адаптивни градијент

⇒ adaptive gradient

$$g_{ki} = \nabla f(x_k)_i = \frac{\partial f(x_k)}{\partial x_i}$$

$$G_{ki} = \sum_{j=0}^{k} g_{ji}^2$$ ⇒ сума градијената до сада

$$\boxed{x_{k+1} = x_{ki} - \frac{\gamma}{\sqrt{G_{ki} + \varepsilon_1}} g_{ki}}$$

Желимо: стрмо → спорије кретање

равно → брже кретање

(−)
↳ иако на почетку функционише, временом неће бити идеално јер гледамо ~~све градијенте~~, што „деградира" понашање

Идеално → последњих неколико итерација, а не све

⑤ ADAM

⇒ најчешће се користи

⇒ прилично адаптивно

$$\left.\begin{aligned} m_k &= w_1 m_k + (1-w_1) g_k \\ v_k &= w_2 v_k + (1-w_2) g_k^2 \end{aligned}\right\} \quad \hat{m}_k = \frac{m_k}{1-w_1}, \quad \hat{v}_k = \frac{v_k}{1-w_2}$$

$$\Rightarrow \boxed{x_{k+1} = x_{k-1} - \frac{\gamma}{\sqrt{v_k + \varepsilon_1}} m_k}$$