Inteligentni Sistemi

- ## Optimizacija

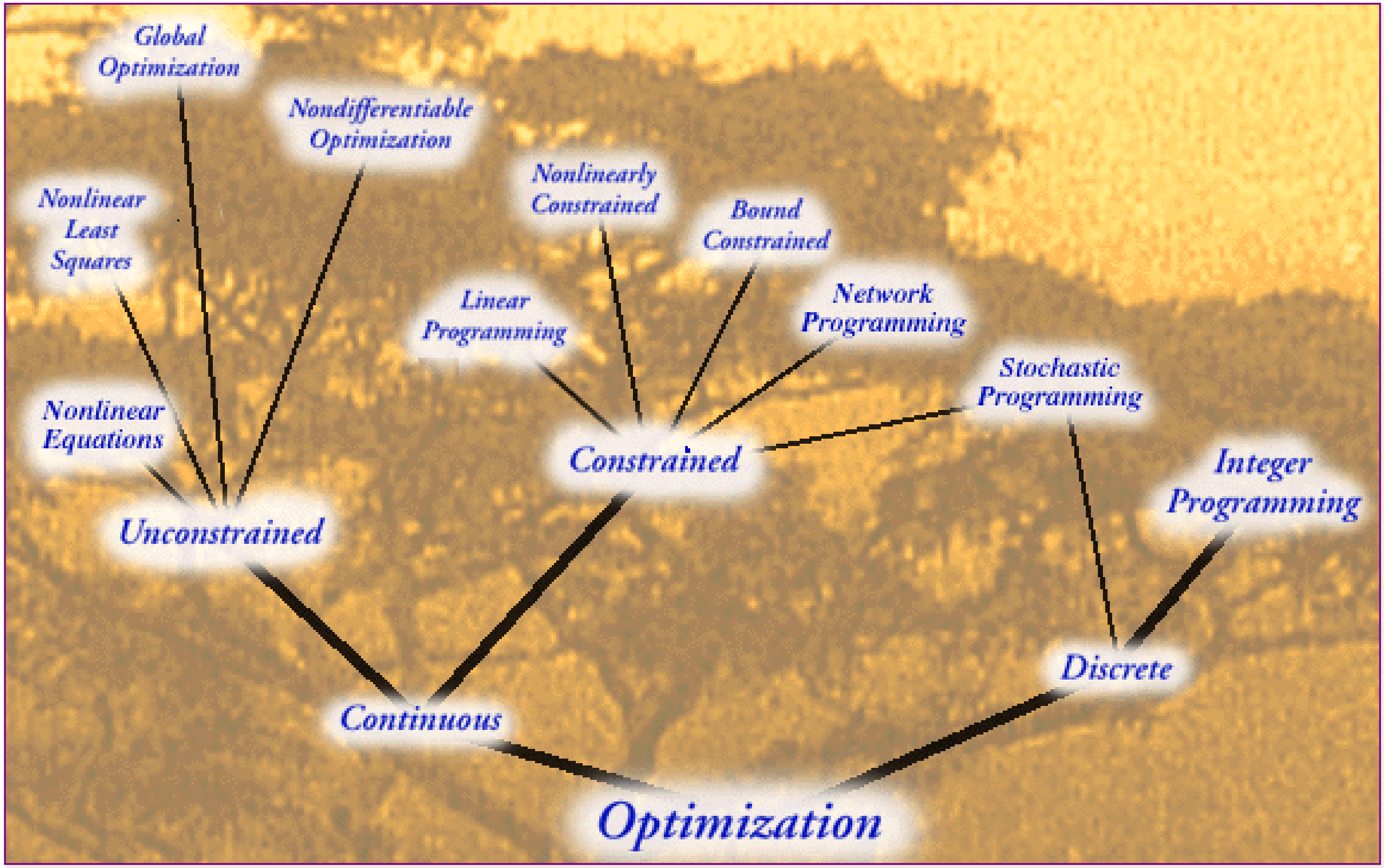
- Prof. Jurij F. Tasič
- asistenta: Emil Plesnik (vaje), Jana Milenković (projekti)

Vsebina

- Ciljna funkcija
- Spremenljivke
- Omejitve

Iskanje vrednosti spremenljivk,
ki minimizirajo ali maksimizirajo ciljno funkcijo,
pri tem pa zadoščajo danim omejitvam

Različne vrste optimizacije



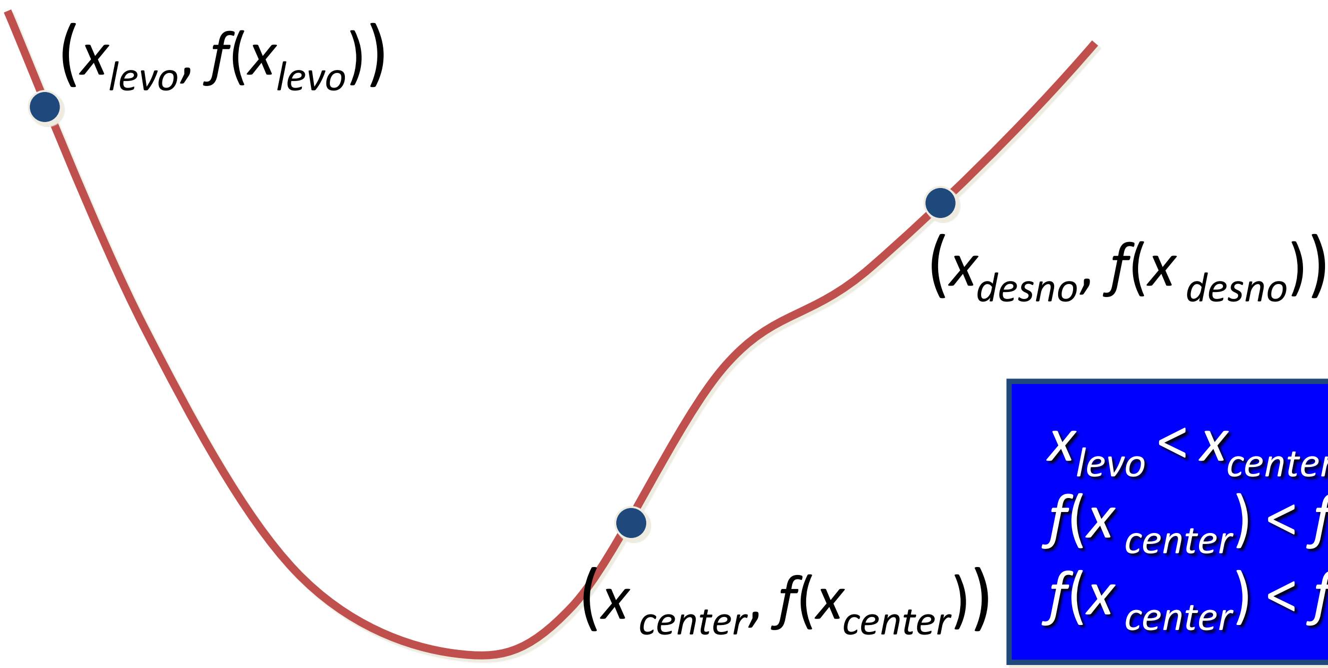
Vir: Optimization Technology Center
http://www-fp.mcs.anl.gov/otc/Guide/OptWeb/

Različne optimizacijske tehnike

- Algoritmi so različni glede na specifični problem
  - Zaprta (analitična) oblika vs. numerični vs. diskretni
  - Lokalni vs. globalni minimumi
  - Računska kompleksnost v obsegu od O(1) do NP-hard
- Danes:
  - Zvezne numerične metode

Optimizacija v 1-D

- Analogija z oklepanjem pri iskanju korenov
- Kaj pomeni *oklepati* minimum?



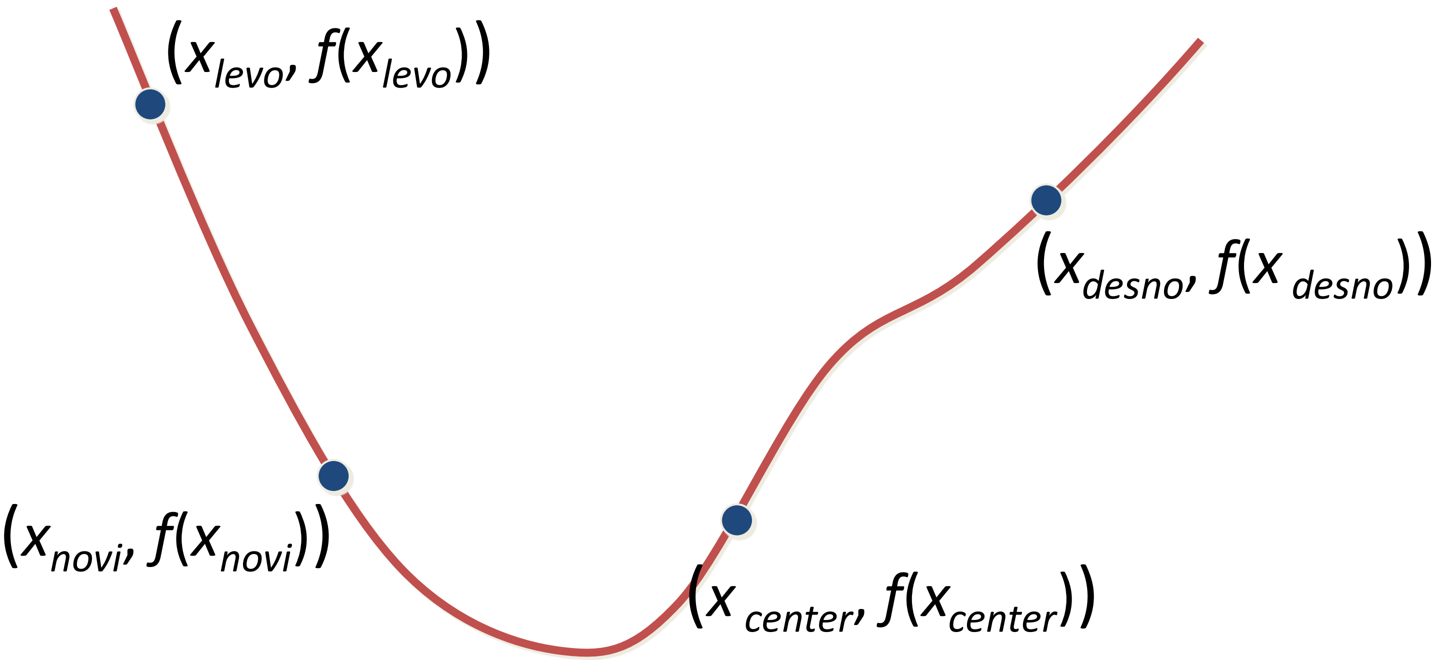
$$x_{levo} < x_{center} < x_{desno}$$
$$f(x_{center}) < f(x_{levo})$$
$$f(x_{center}) < f(x_{desno})$$

Optimizacija v 1-D

- Ko so navedene lastnosti znane, je vsaj en lokalni minimum med x_{levo} in x_{desno}
- Začetna postavitev oklepanja:
  - Glede na $x_{začetni}$, *korak*
  - Oceni $f(x_{začetni})$, $f(x_{začetni} + korak)$
  - If vrednost zmanjšuje, se pomikaj s korakom dokler se trend ne obrne
  - Else se pomikaj s korakom v nasprotno smer dokler se trend ne obrne
  - Povečaj korak po vsaki ponovitvi
- Maksismum: zamenjaj f z $-f$

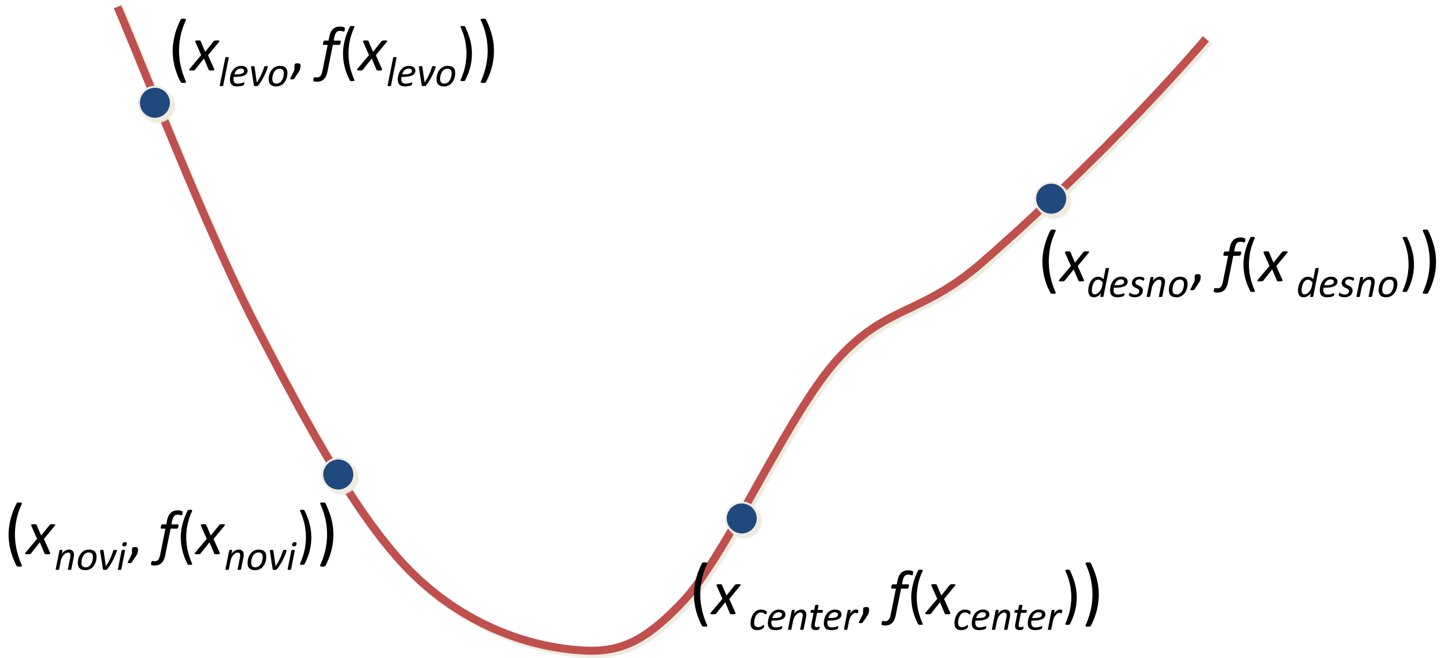
Optimizacija v 1-D

- Strategija: oceni funkcijo pri nekekm x_{novi}



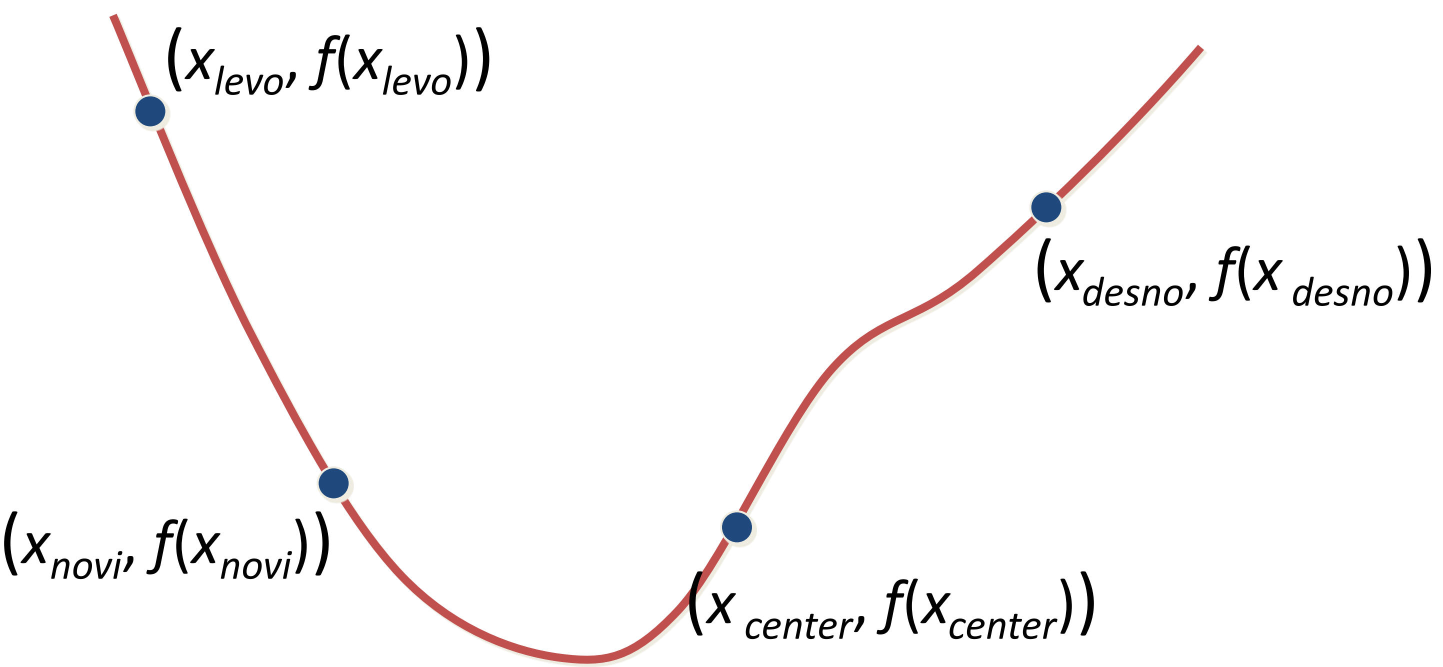
Optimizacija v 1-D

- Strategija: oceni funkcijo pri nekekm x_{novi}
  - Sedaj so nove "oklepne" točke x_{novi}, x_{center}, x_{desno}



Optimizacija v 1-D

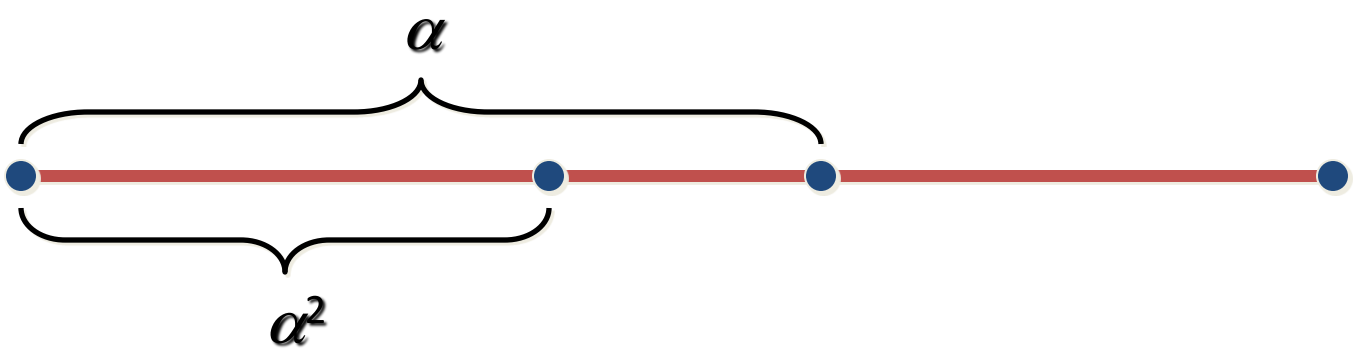
- Strategija: oceni funkcijo pri nekekm x_{novi}
  - Sedaj so nove "oklepne" točke x_{levo}, x_{novi}, x_{center}



Optimizacija v 1-D

- Nasprotno kot pri iskanju korenov ni možno zagotoviti, da se bo interval po vsakem koraku zmanjšal za faktor 2
- Poiščimo optimalno lokacijo za x_{center} v odvisnosti od leve in desne točke, ki bo zagotavljala enak faktor zmanjševanja ne glede na izzid

Optimizacija v 1-D



if $f(x_{novi}) < f(x_{center})$

novi interval = α

else

novi interval = $1-\alpha^2$

Metoda zlatega reza

- Da zagotovimo enak interval, postavimo pogoj $\alpha = 1-\alpha^2$
- Da velja, $\alpha = \frac{\sqrt{5}-1}{2} = \overline{\varphi}$
- To je fatktor "zlatega reza" = 0.618...
- Interval se tako zmanjša za 30% na iteracijo
  - *Linearna konvergenca*

Toleranca napake

- V bližini minimuma je odvod = 0, zato velja

$$f(x+\Delta x) = f(x) + \tfrac{1}{2} f''(x)\Delta x^2 + \ldots$$

$$f(x+\Delta x) - f(x) = \tfrac{1}{2} f''(x)\Delta x^2 = \text{machine } \varepsilon$$

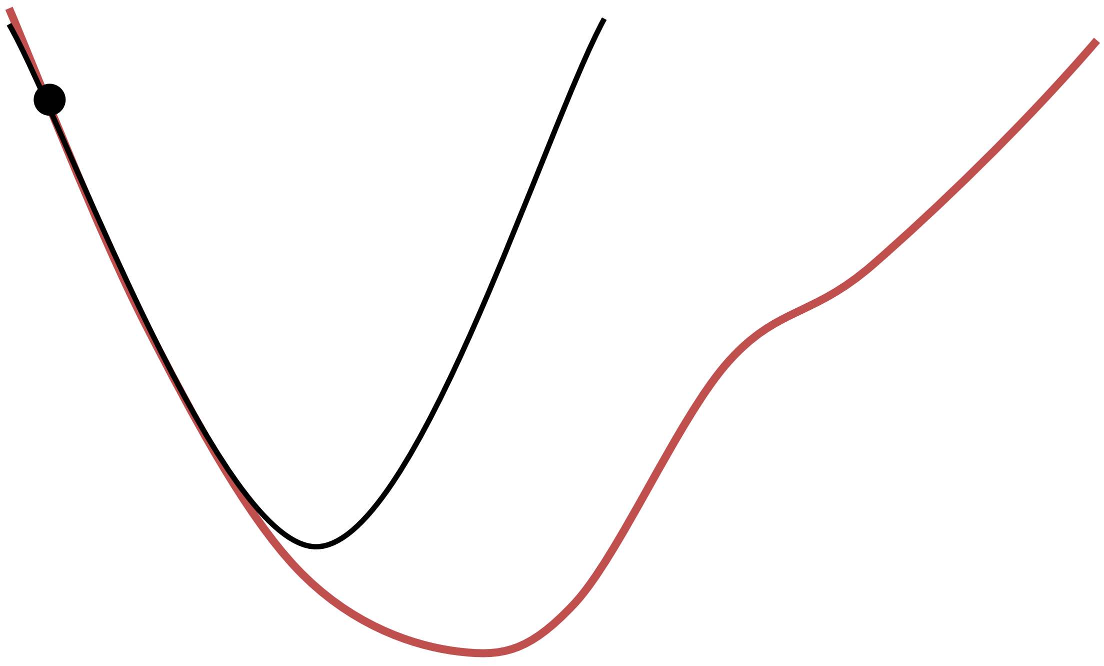
$$\Rightarrow \Delta x \sim \sqrt{\varepsilon}$$

- Pravilo palca: nesmiselno je iskati večjo natančnost, kot sqrt(ε)
  - Uporaba double precision, če želimo single-precision rezultat (in/ali imamo single-precision podatke)

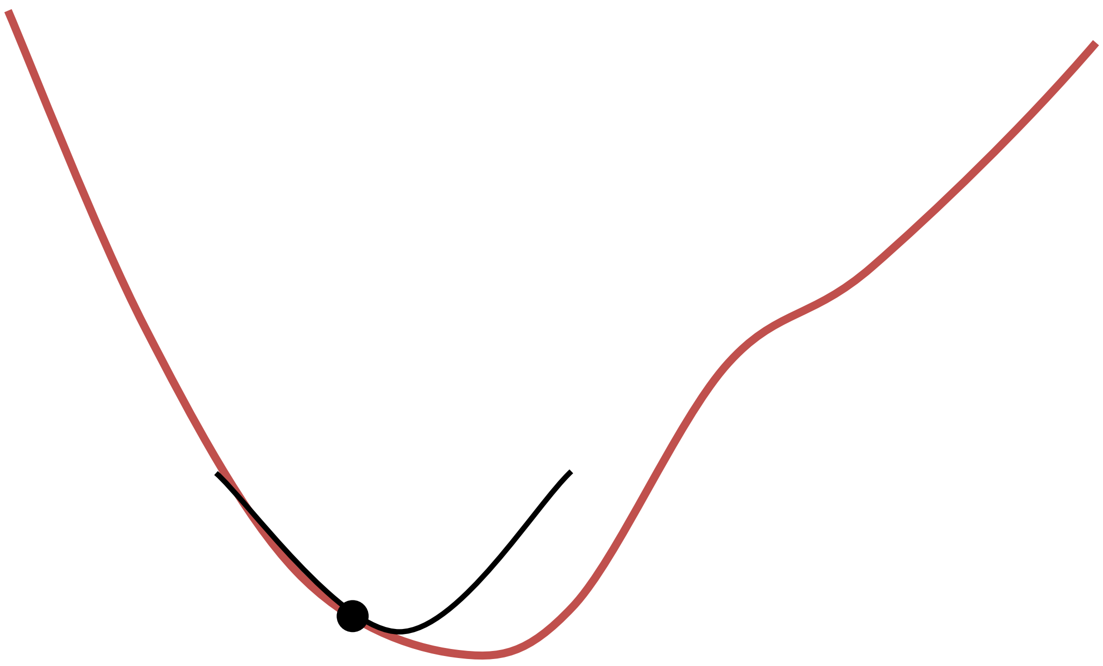
Hitrejša 1-D optimizacija

- Kompromis med super in linearno konvergenco za manjšo robustnost
  - Kombinacija z Zlatim rezom za varnost
- Uporabni napotki:
  - Skozi 3 točke položimo parabolo in poiščemo minimum
  - Izračunamo odvode in položaje, uporabimo kubično prilagajanje
  - Uporabimo druge odvode: Newton

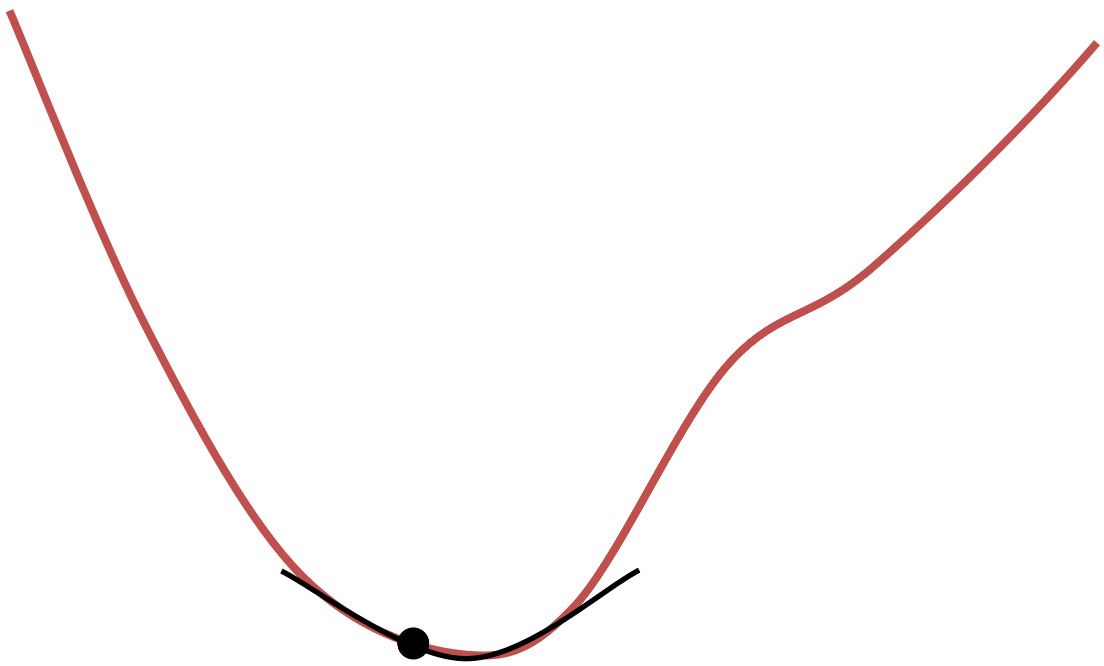
Newton-ova metoda



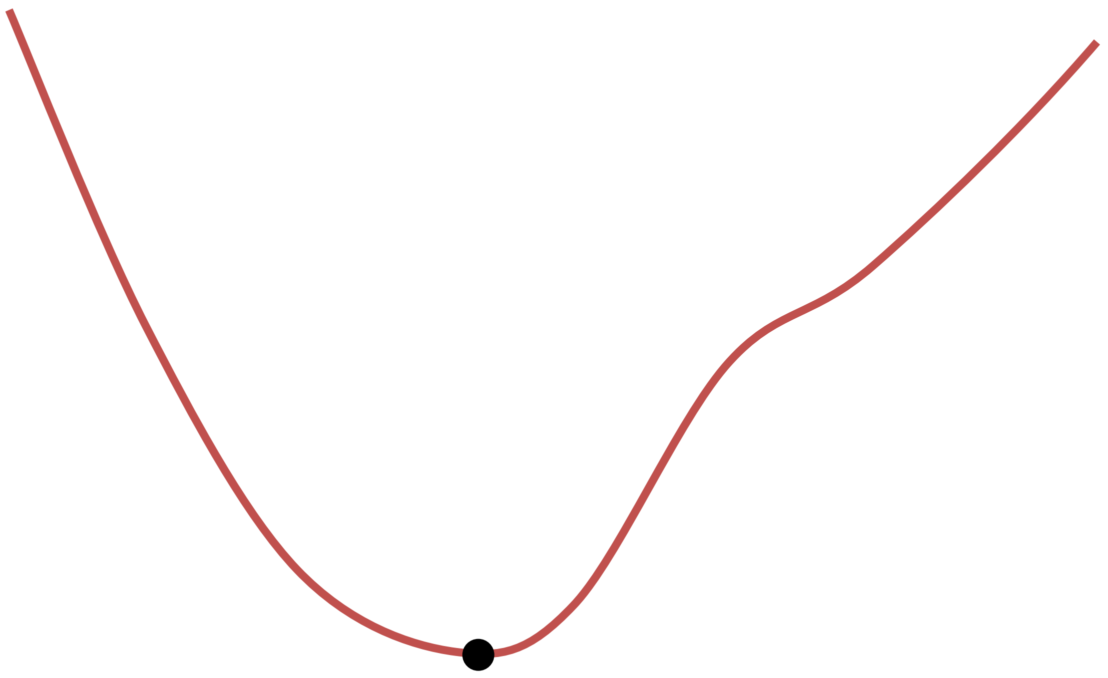
Newton-ova metoda



Newton-ova metoda



Newton-ova metoda



Newton-ova metoda

- Pri vsakem koraku:

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

- potrebujemo 1. in 2. odvod
- Kvadratična konvergenca

Več-dimenzionalna optimizacija

- Pomembna na več področjih
  - Prilagajanje modela izmerjenim podatkom
  - Iskanje najboljšega načrta v nekem prostoru parametrov
- V splošnem zahtevna
  - Nenavadne oblike: več ekstremov, sedel, neenakomernih potekov „dolin“, itd.
  - Oklepanje ni možno
- Enostavnejša od iskanja korenov
  - Zmeraj obstaja pot “navzdol”

Newton-ova metoda v več dimenzijah

- Nadomestimo 1. odvod z gradientom, 2. odvod pa s Hessovo matriko

$$f(x,y)$$

$$\nabla f = \begin{pmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{pmatrix}$$

$$H = \begin{pmatrix} \dfrac{\partial^2 f}{\partial x^2} & \dfrac{\partial^2 f}{\partial x \partial y} \\ \dfrac{\partial^2 f}{\partial x \partial y} & \dfrac{\partial^2 f}{\partial y^2} \end{pmatrix}$$

Newton-ova metoda v več dimenzijah

- Nadomestimo 1. odvod z gradientom, 2. odvod pa s Hessovo matriko
- Velja

$$\vec{x}_{k+1} = \vec{x}_k - H^{-1}(\vec{x}_k)\nabla f(\vec{x}_k)$$

- Zelo občutljivo, razen če je funkcija zelo gladka in začenjamo blizu minimuma

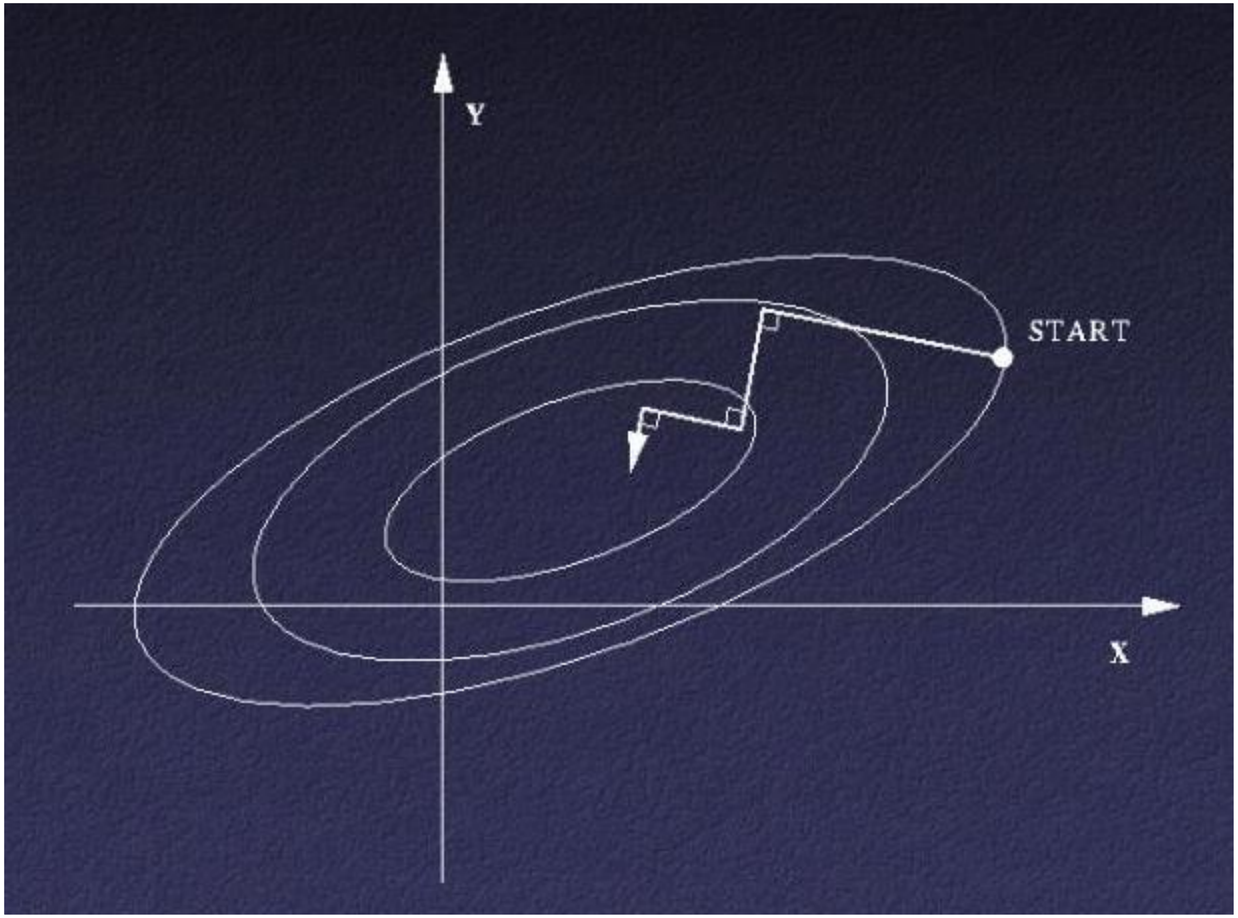
Klasifikacija metod

- Uporaba funkcija + gradient + Hessova matrika (Newton)
- Uporaba funkcija + gradient (večina descent metod)
- Uporaba funkcijskih vrednosti (Nelder-Mead, imenovana tudi “simplex” metoda)

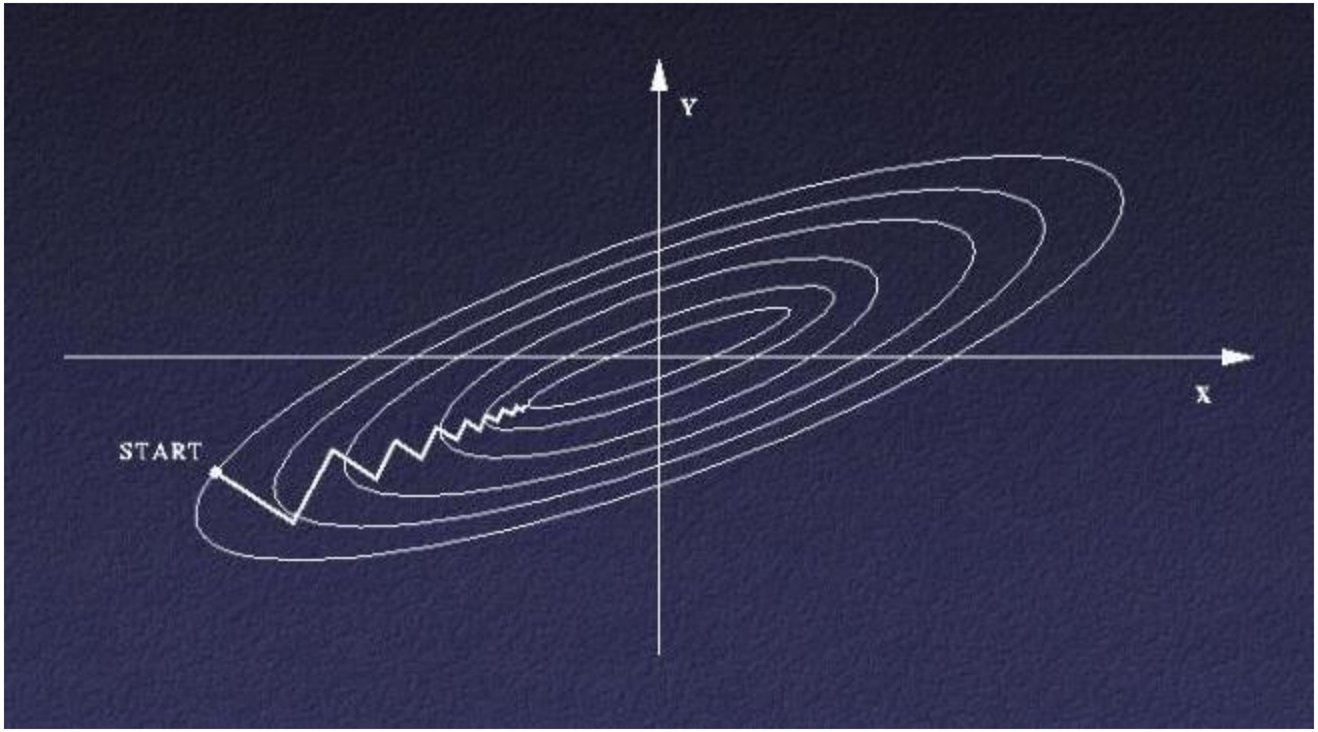
Metoda najstrmejšega zmanjševanja (Steepest descent)

- Kaj če ne moremo/ne želimo uporabiti 2. odvoda?
- “Kvazi-Newtonove” metode uporabijo približek Hessove matrike
- Alternativno: pomikanje ob (negativnem) gradientu…
  - Izvedba 1-D minimizacije vzdolž prehoda skozi trenutno točko v smeri gradienta
  - Potem ponoven izračun gradienta, nato iteracije

Slabost najstrmejšega zmanjševanja



Slabost najstrmejšega zmanjševanja



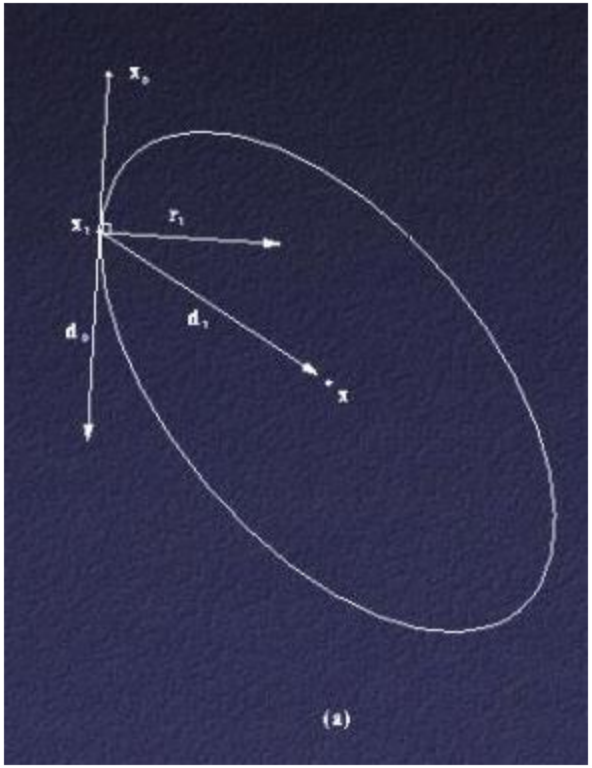
Metoda konjugiranega gradienta

- Ideja: izogniti se „razveljavitvi“ minimizacije, ki je že bila opravljena
- Pomikanje v smeri

$$d_{k+1} = -g_{k+1} + \beta_k d_k$$

- Polak Ribiere-ova enačba:

$$\beta_k = \frac{g_{k+1}^{\mathrm{T}} (g_{k+1} - g_k)}{g_k^{\mathrm{T}} g_k}$$



Metoda konjugiranega gradienta

- Konjugirani gradient implicitno pridobi informacijo o Hessovi matriki
- Za kvadratno funkcijo v *n* dimenzijah pridobi natančno rešitev v *n* korakih (ob neupoštevanju napake zaokroževanja)
- Deluje dobro v praksi…

Metode vrednotenja v več dimenzijah

- Če ni mogoče ovrednotiti gradienta, imamo težave
- Lahko uporabimo približne (numerično ocenjene) gradiente:

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f}{\partial e_1} \\ \frac{\partial f}{\partial e_2} \\ \frac{\partial f}{\partial e_3} \\ \vdots \end{pmatrix} \approx \begin{pmatrix} \frac{f(x+\delta \cdot e_1) - f(x)}{\delta} \\ \frac{f(x+\delta \cdot e_2) - f(x)}{\delta} \\ \frac{f(x+\delta \cdot e_3) - f(x)}{\delta} \\ \vdots \end{pmatrix}$$

Strategije generične optimizacije

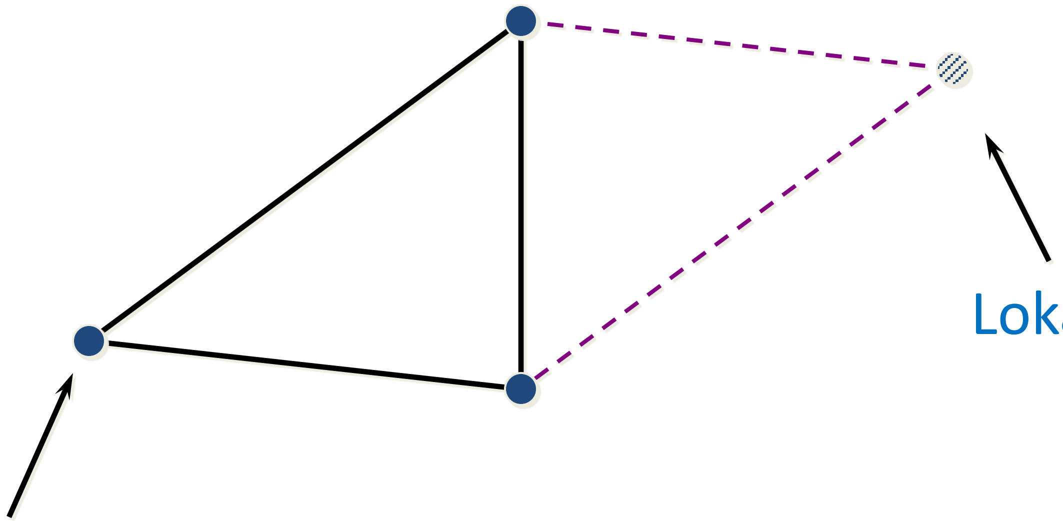
- Enotno vzorčenje:
  - Zahtevnost narašča eksponentno s številom dimenzij
- Simulirano žarjenje:
  - Iskanje v naključnih smereh
  - Na začetku z velikimi koraki, ki jih postopno zmanjšujemo
  - "Razpored žarjenja" – kako hitro upočasniti?

Simplex metoda (Nelder-Mead)

- Sledimo *n*+1 točkam v *n* dimenzijah
  - Vozlišča *simplex* (trikotnik v 2D tetraeder v 3D, itd.)
- Pri vsaki iteraciji: simplex se lahko premakne, razširi ali skrči
  - Poznana tudi kot *ameba metoda*: simplex "izžareva" vzdolž funkcije

Simplex metoda (Nelder-Mead)

- Osnovna operacija: <u>zrcaljenje</u>

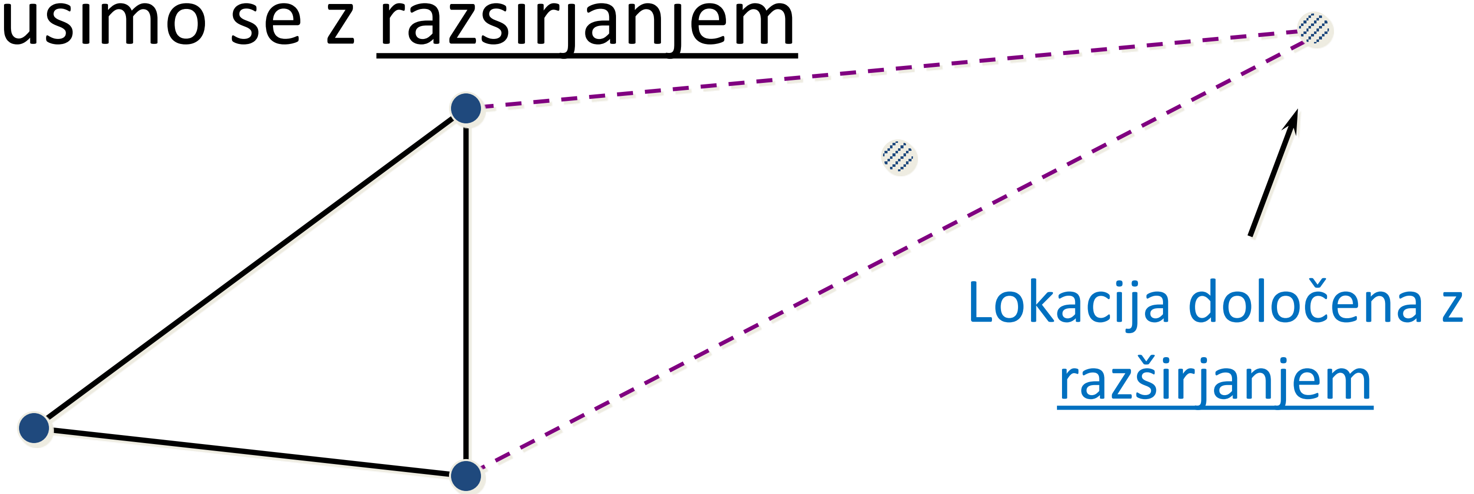


Lokacija določena z <u>zrcaljenjem</u>

Najslabša točka
(največja funkcijska vrednost)

Simplex metoda (Nelder-Mead)

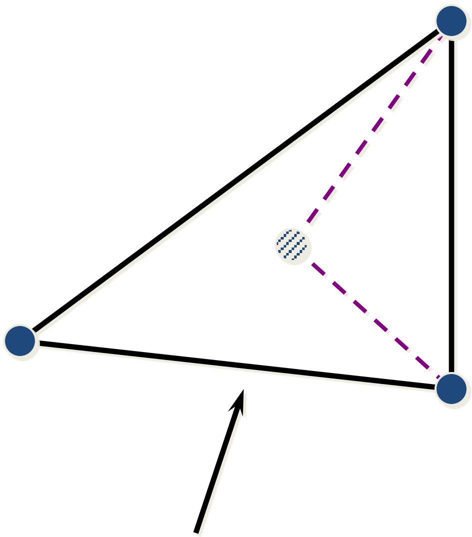
- Če je zrcaljenje določilo najboljšo (najmanjšo) vrednost do sedaj, potem poskusimo še z razširjanjem



- V nasprotnem primeru, če je zrcaljenje učinkovalo v kakršni koli meri, ga nadaljujemo

Simplex metoda (Nelder-Mead)

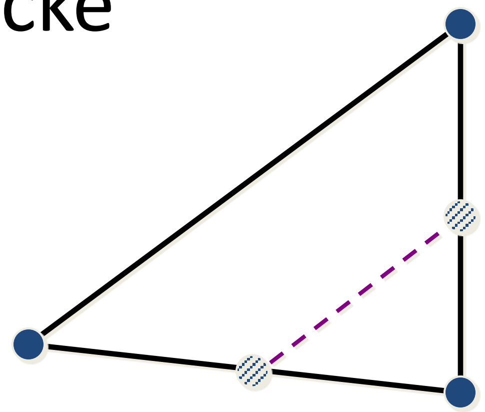
- Če zrcaljenje ni učinkovalo (zrcaljena točka je še vedno najslabša) poskusimo s <u>krčenjem</u>



Lokacija določena s <u>krčenjem</u>

Simplex metoda (Nelder-Mead)

- Če ni delovala nobena od prej omenjenih operacij, zmanjšamo simplex okrog *najboljše* točke



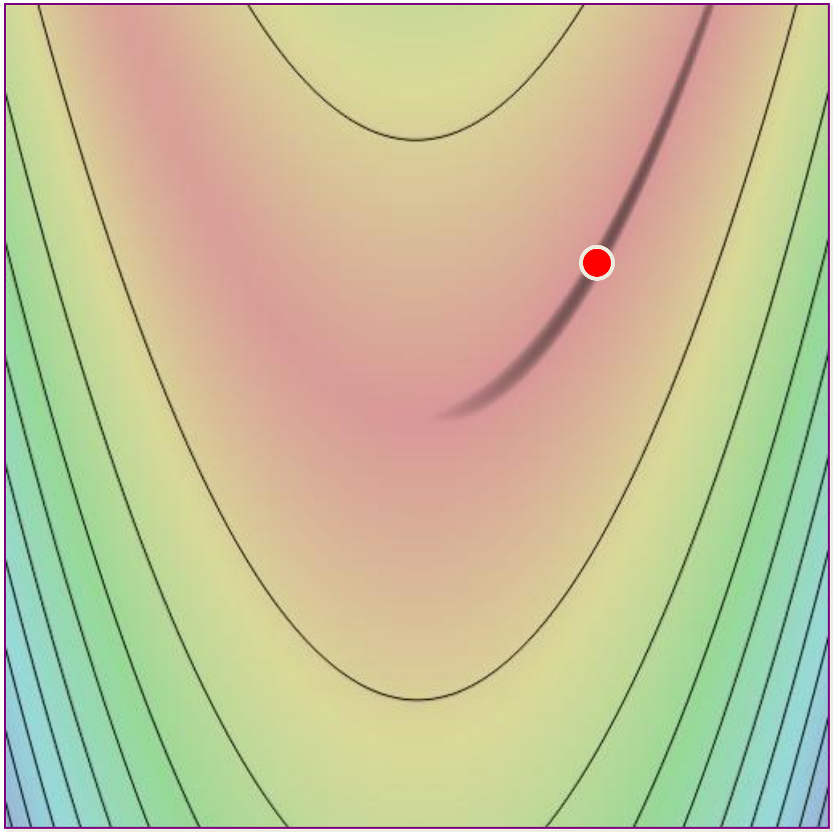
Simplex metoda (Nelder-Mead)

- Metoda je precej učinkovita ob vsaki iteraciji (tipično 1 – 2 oceni funkcije)
- Lahko zahteva *veliko* iteracij
- Nekoliko nezanesljiva – včasih je potreben *ponoven začetek* zaradi sesedanja simplex-a
- Prednosti: preprosta za implementacijo, ne potrebuje odvoda, ni odvisna od gladkosti funkcije itd.

Rosenbrock-ova funkcija

- Oblikovana posebej za preizkušanje optimizacijskih tehnik
- Ukrivljena, ozka dolina

$$f(x, y) = 100(y - x^2)^2 + (1 - x)^2$$



Pogojena optimizacija

- Pogoj enakosti: optimiziraj *f(x)* glede na $g_i(x)=0$
- Metoda Lagrange-ovih multiplikatorjev: preslikava v več-dimenzionalni problem $f(x)+\sum \lambda_i g_i(x)$
- Minimiziraj $(x_1 \ldots x_n; \lambda_1 \ldots \lambda_k)$

  glede na

Pogojena optimizacija

- Pogoji neenakosti so zahtevnejši…
- Če so ciljna funkcija in pogoji linearni, potem gre za „linearno programiranje“
- Opomba: minimum mora biti na robu regije, ki jo oblikujejo pogoji
- Simplex metoda: med premikanjem od vozlišča do vozlišča minimiziramo ciljno funkcijo

Pogojena optimizacija

- V splošnem je “nelinearno programiranje” zahtevno
- Algoritmi za posebne primere (npr. kvadratni)

Globalna optimizacija

- V splošnem ne moremo zagotoviti, da smo našli globalni minimum in ne samo lokalnega
- Nekaj hevrističnih tehnik:
  - Multi-start: poskusimo z lokalno optimizacija iz več različnih začetnih točk
  - Zelo počasno simulirano žarjenje
  - Uporaba analitičnih metod za določanje obnašanja, vodenja metod v prava območja

Tehnike numerične optimizacije

Tehnike nepogojene več-parameterske optimizacije

Direktno iskanje (brez uporabe informacij o odvodih):

- Hooke-Jeeves-ovo iskanje vzorcev
- Nelder-Mead-ova serijska simplex metoda
- Powell-ova metoda konjugiranih smeri
- Različne evolucijske tehnike

Tehnike numerične optimizacije

Tehnike nepogojene več-parameterske optimizacije

Gradientne metode (zahtevajo uporabo odvoda):

- Najstrmejše zmanjševanje
- Fletcher-Reeves-ova metoda konjugiranega gradienta

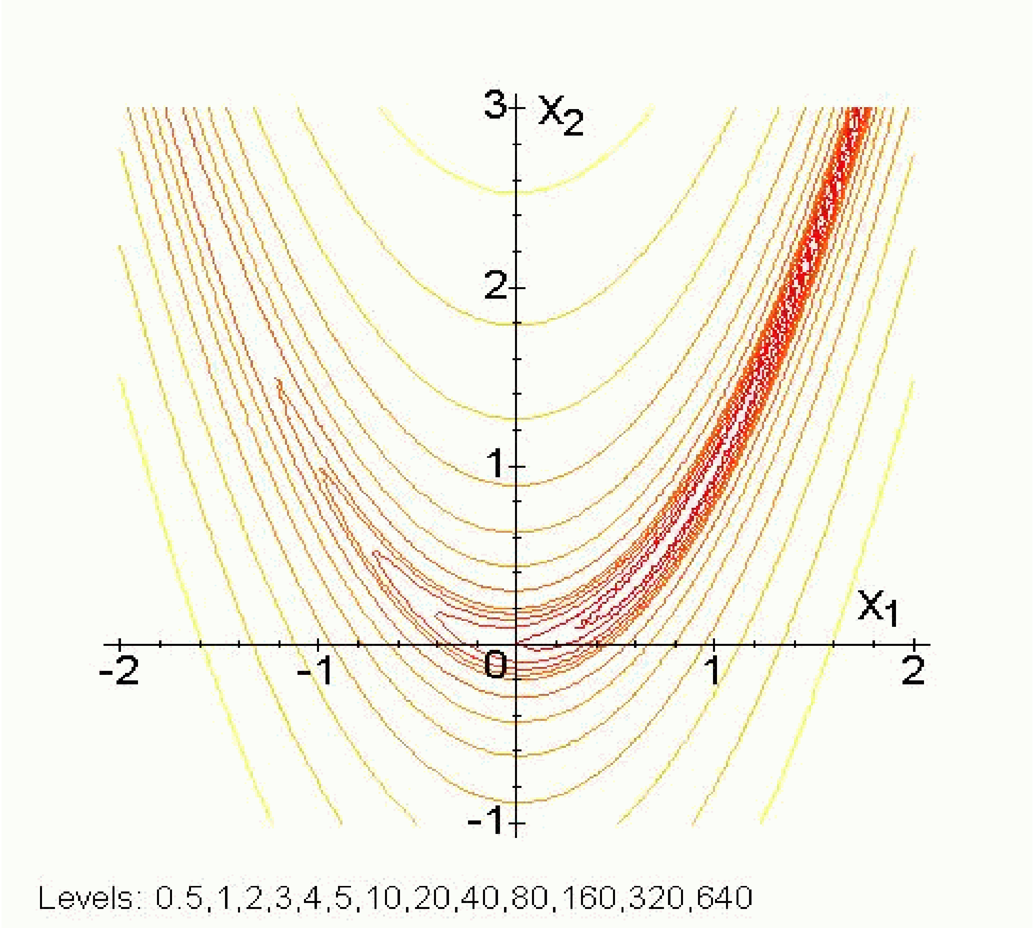
Metode drugega reda (zahtevana je uporaba drugega odvoda):

- Newton-ova metoda
- Kvazi-Newton-ova metoda (uporabi približek matrike drugih odvodov)

Tehnike numerične optimizacije
Primer 1

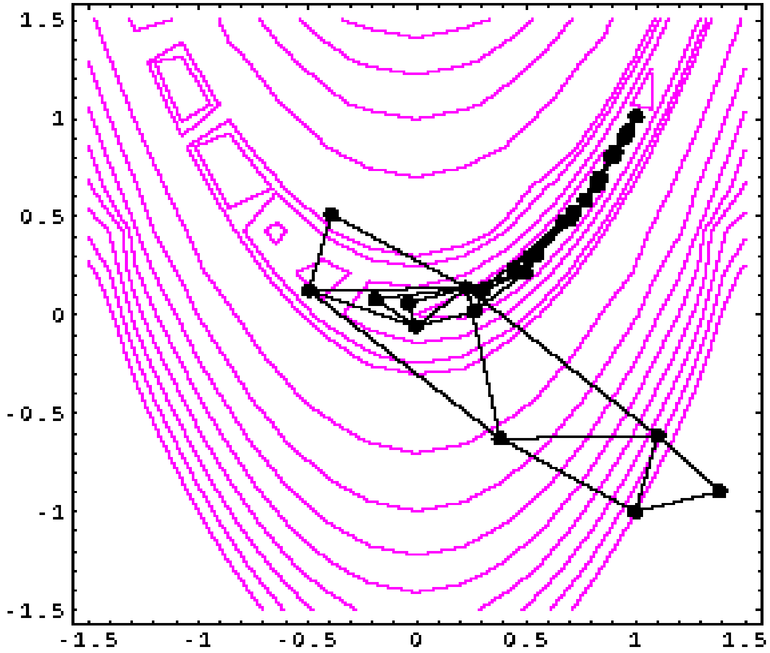
Tehnike nepogojene več-parameterske optimizacije.

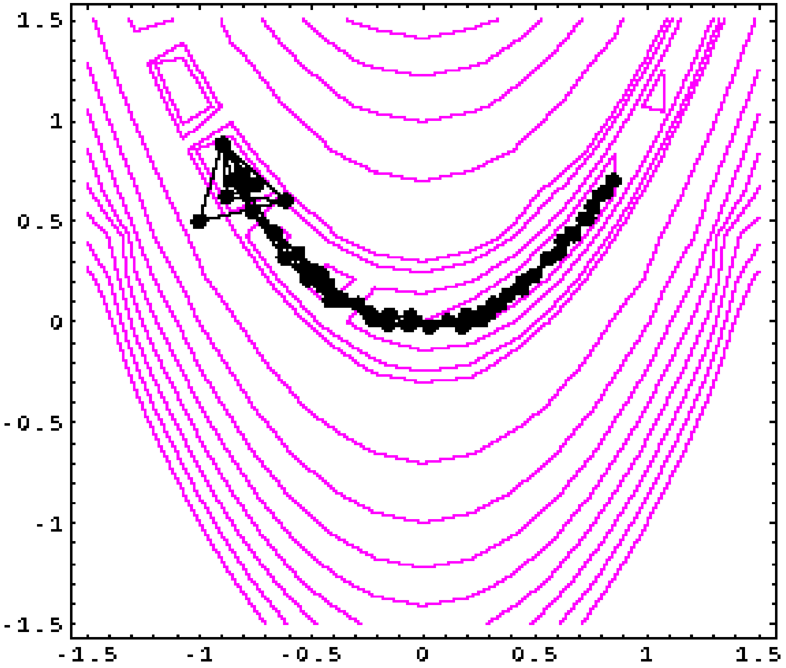
Rosenbrock-ova “banana” funkcija
$F=100(x_2-x_1^2)^2+(x_1-1)^2$



Tehnike numerične optimizacije
Primer 1

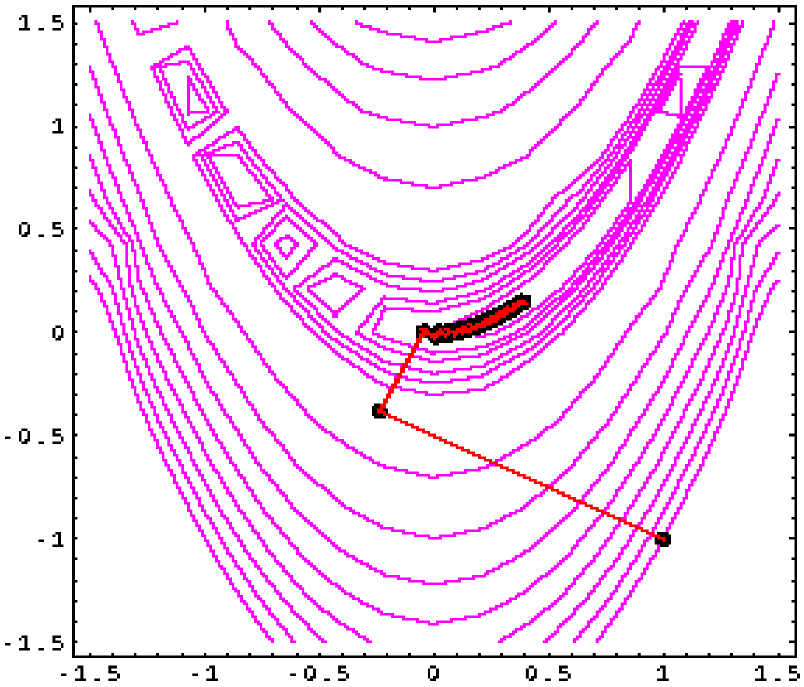
Nelder-Mead-ova serijska simplex metoda

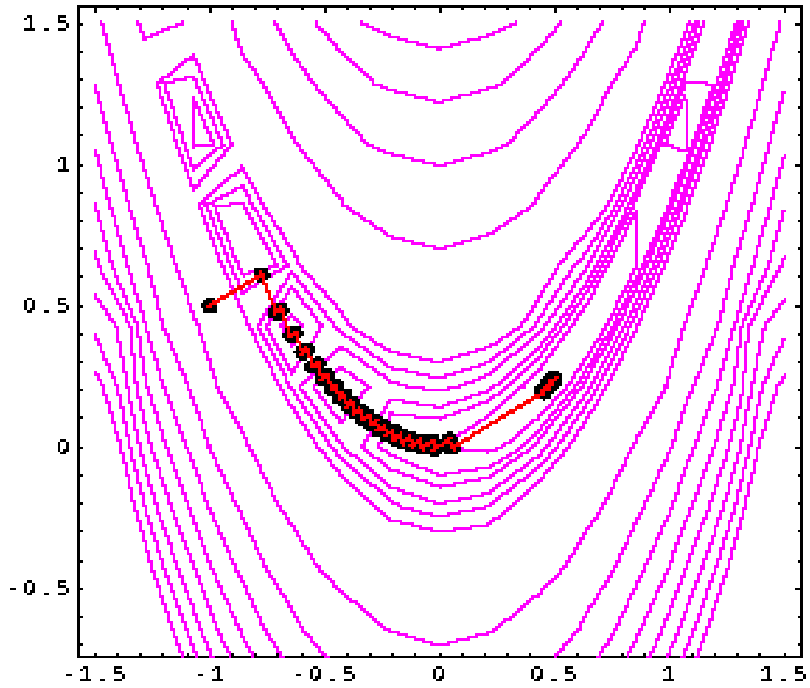




Tehnike numerične optimizacije
Primer 1

Najstrmejše zmanjševanje

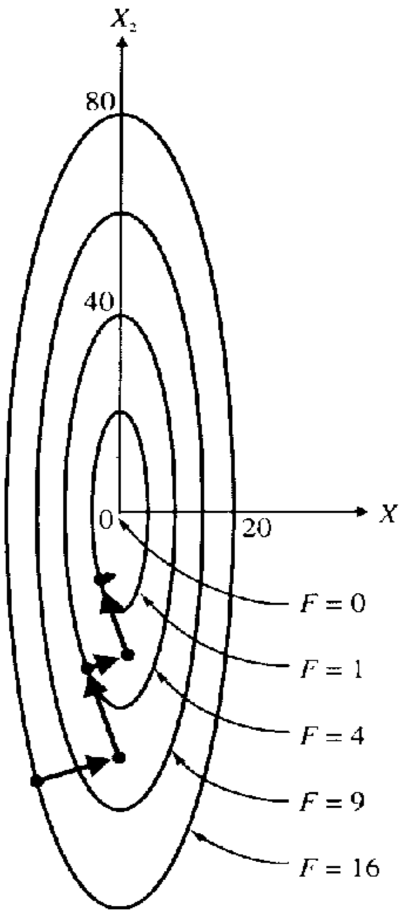




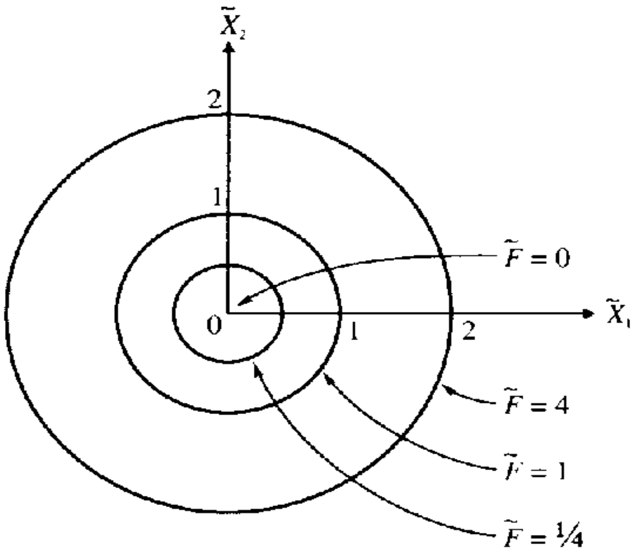
Optimizacijske tehnike

Učinek skaliranja

Zgodovina iteracij



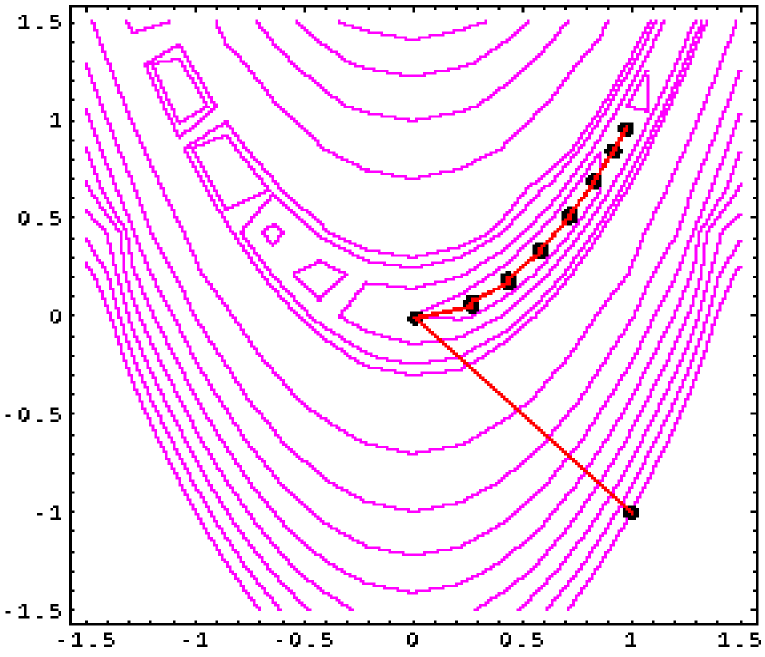
Neskalirano

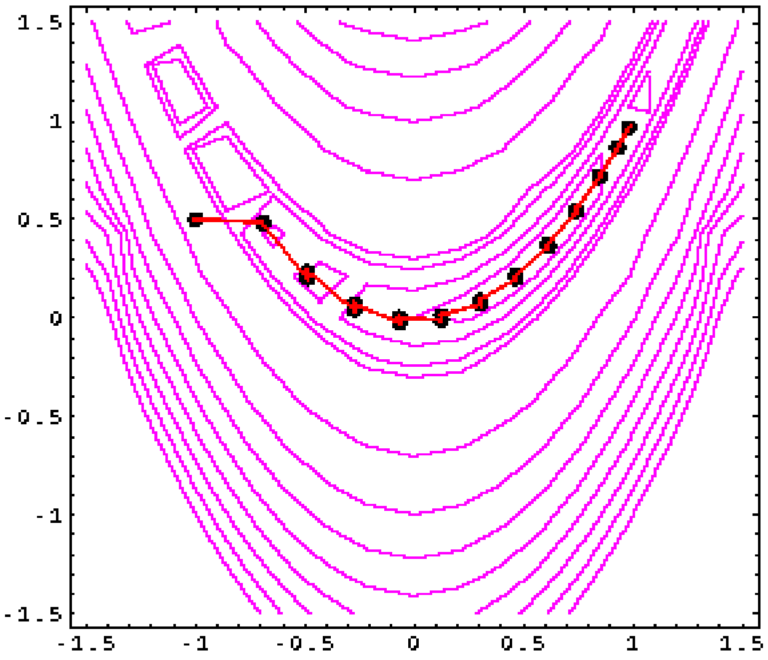


Skalirano

Tehnike numerične optimizacije
Primer 1

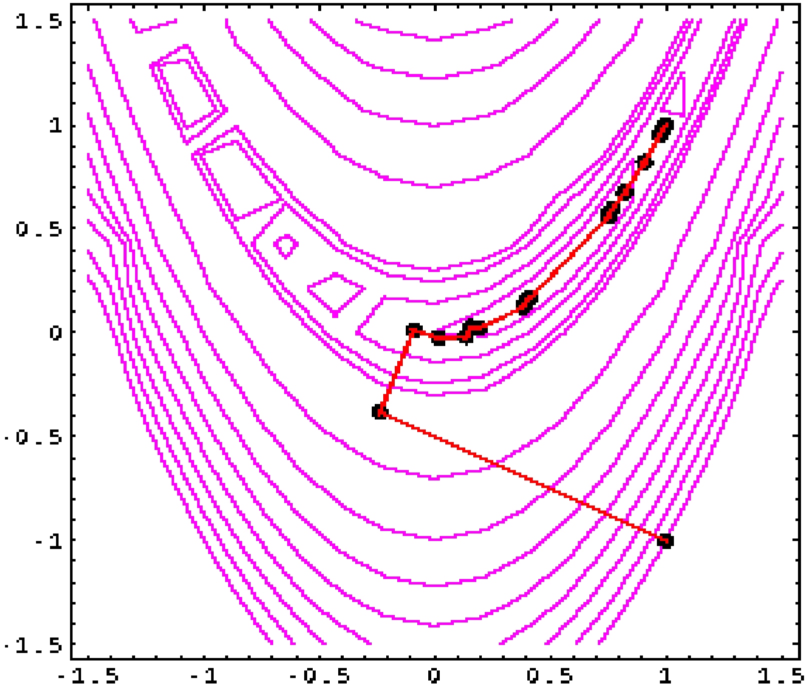
Powell-ova metoda konjugiranih smeri

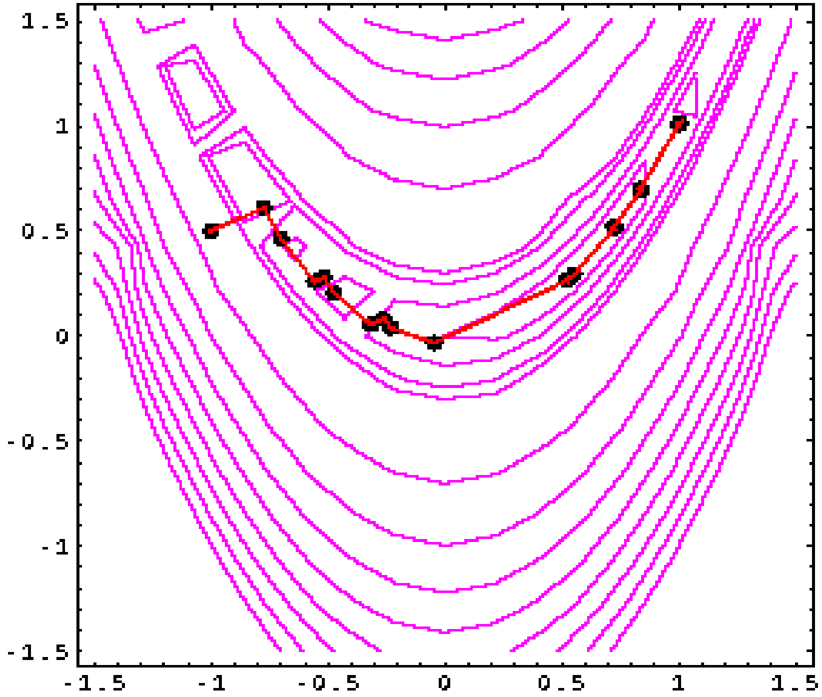




Tehnike numerične optimizacije
Primer 1

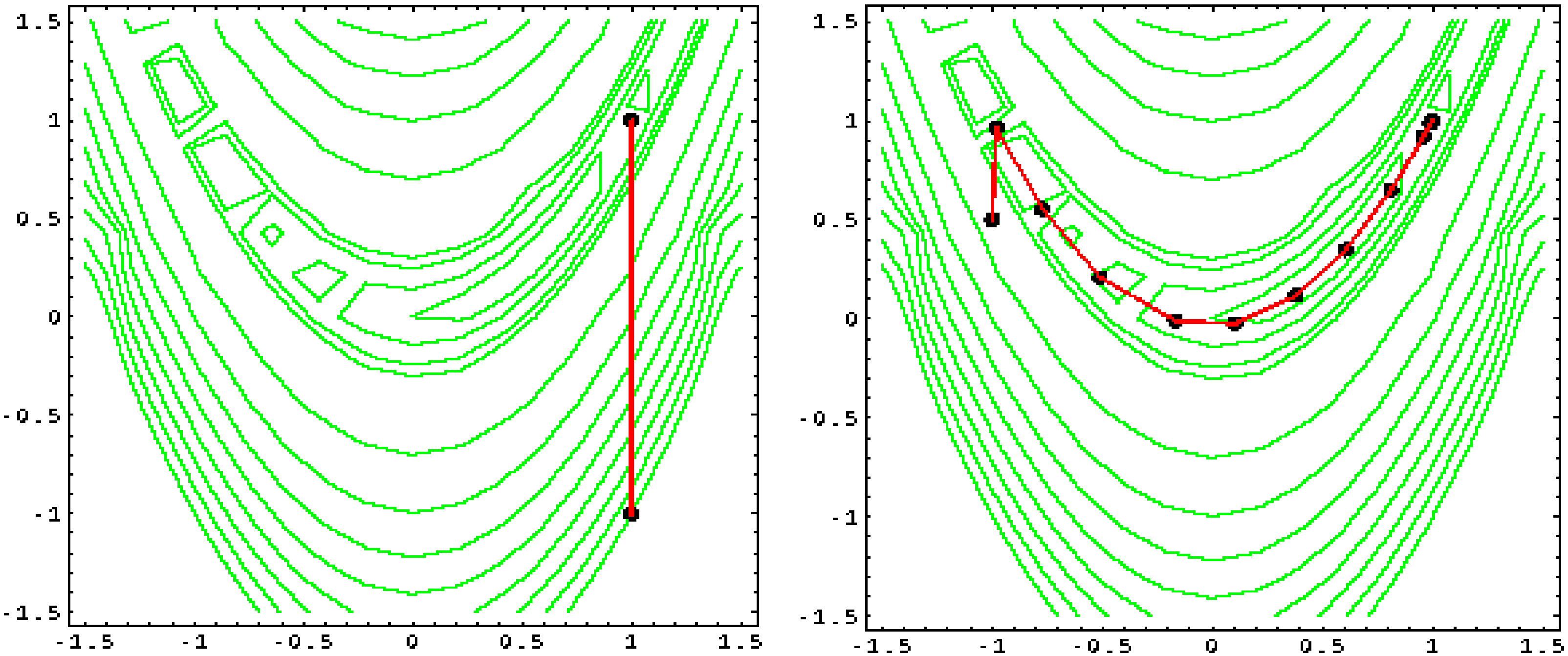
Fletcher-Reeves-ova metoda konjugiranega gradienta





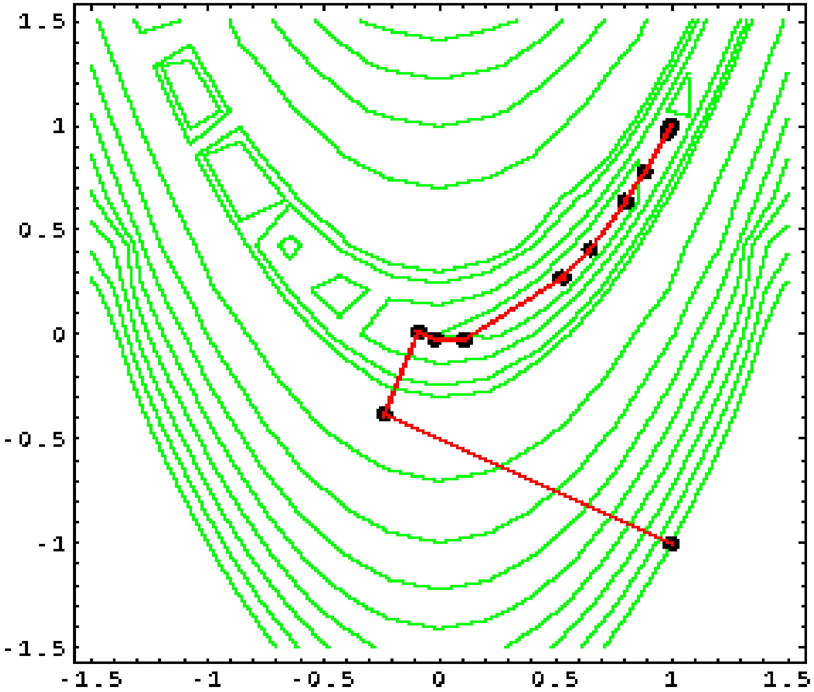
Tehnike numerične optimizacije
Primer 1

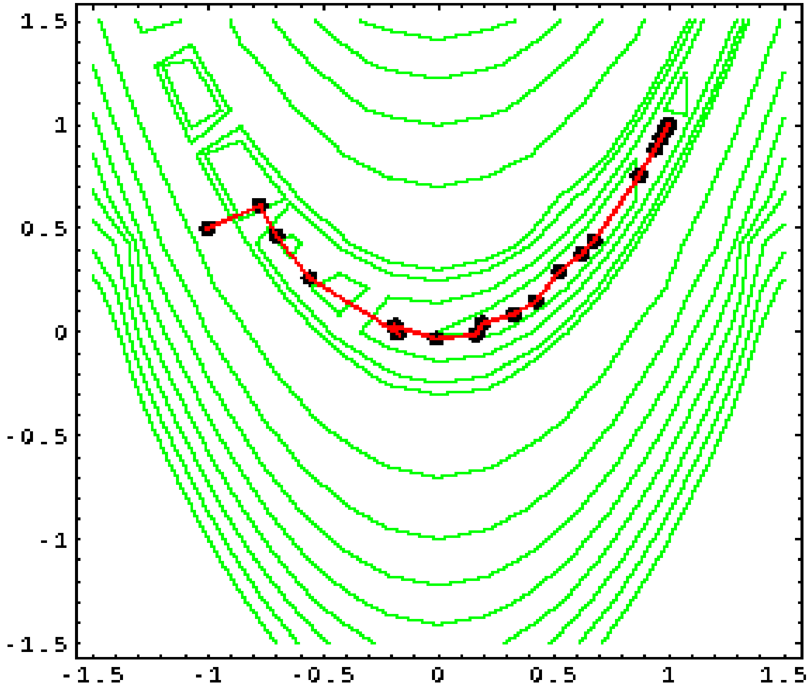
Newton-ova metoda



Tehnike numerične optimizacije
Primer 1

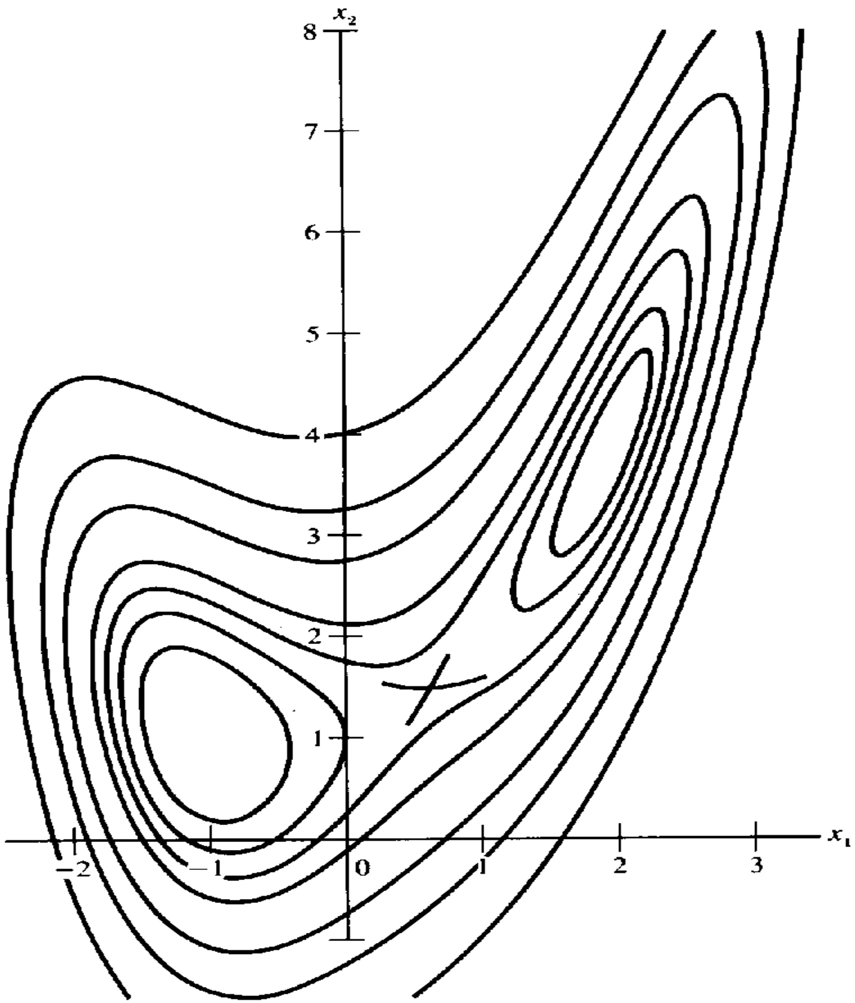
Kvazi-Newton-ova metoda





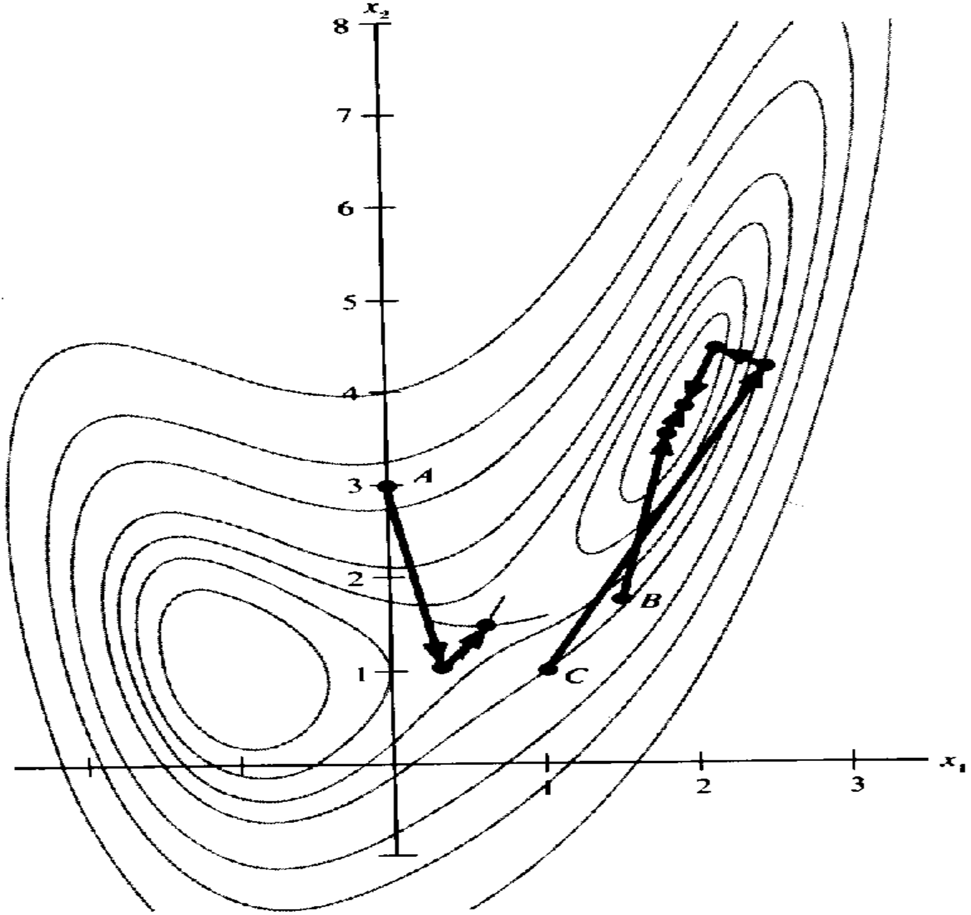
Tehnike numerične optimizacije

Tehnike nepogojene več-parameterske optimizacije. Primer 2.



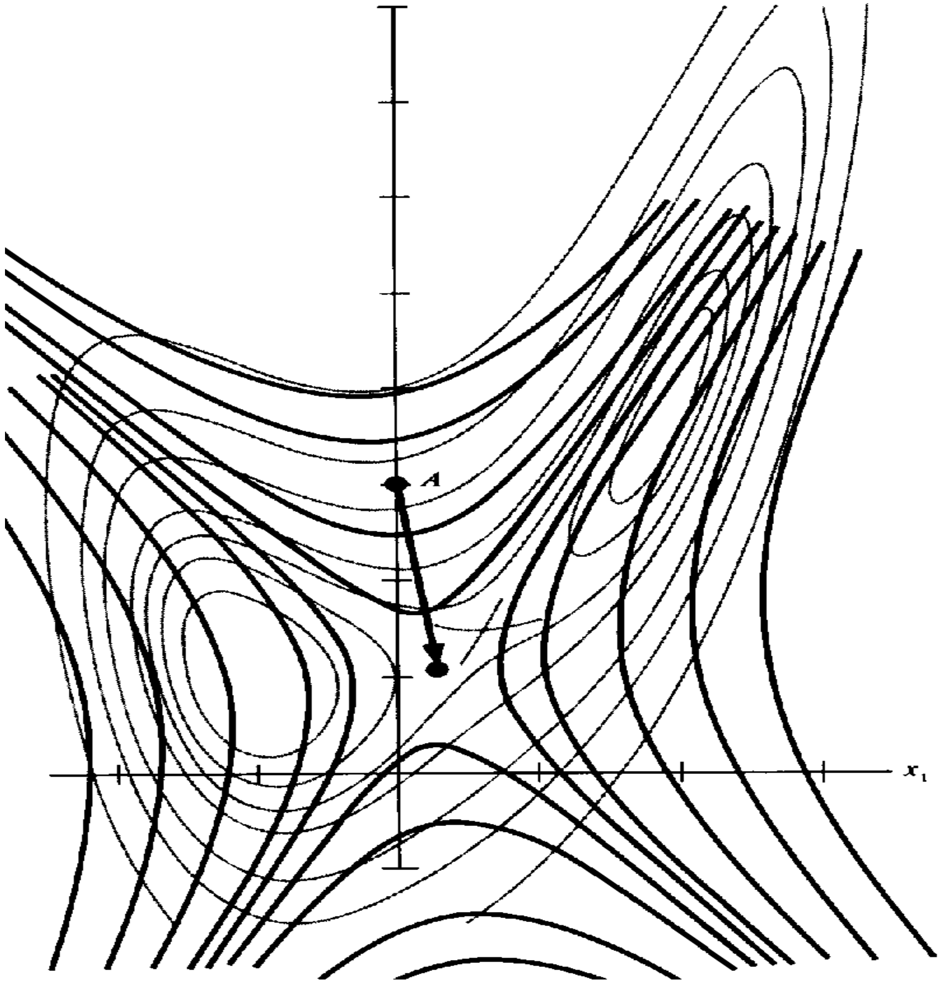
Tehnike numerične optimizacije

Primer 2. Newton, trajektorije iskanja iz treh različnih začetnih točk



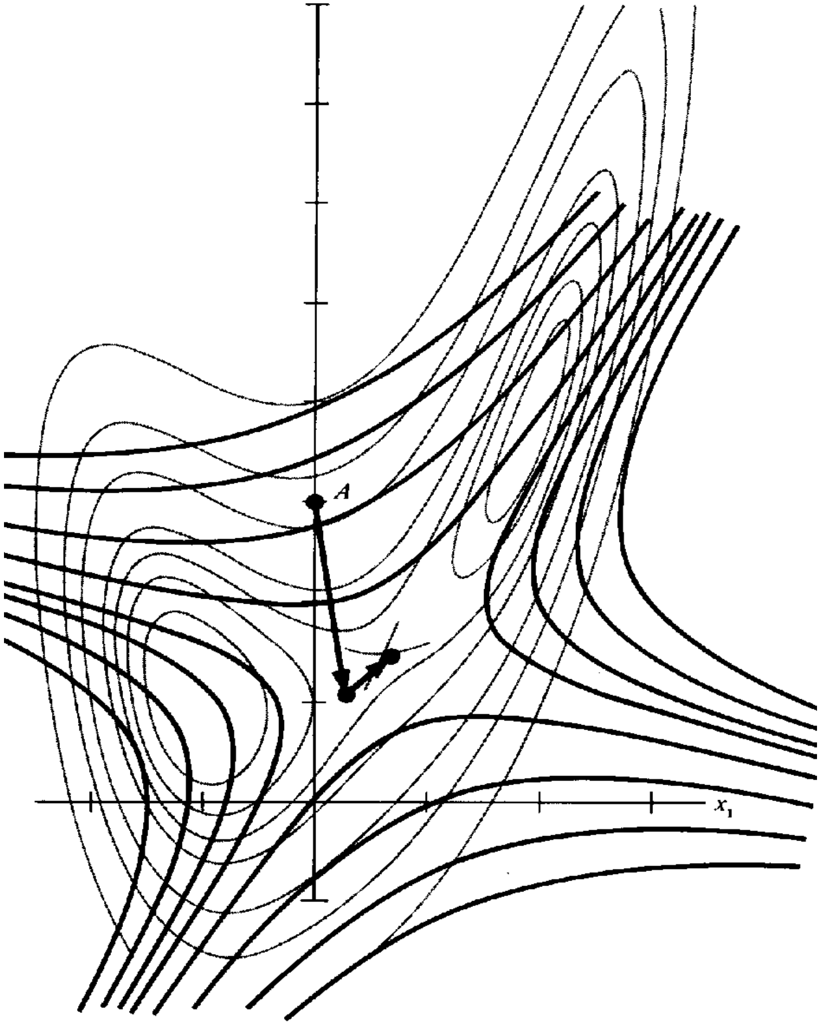
Tehnike numerične optimizacije

Primer 2. Newton, trajektorija iskanja iz prve začetne točke. Prva iteracija.



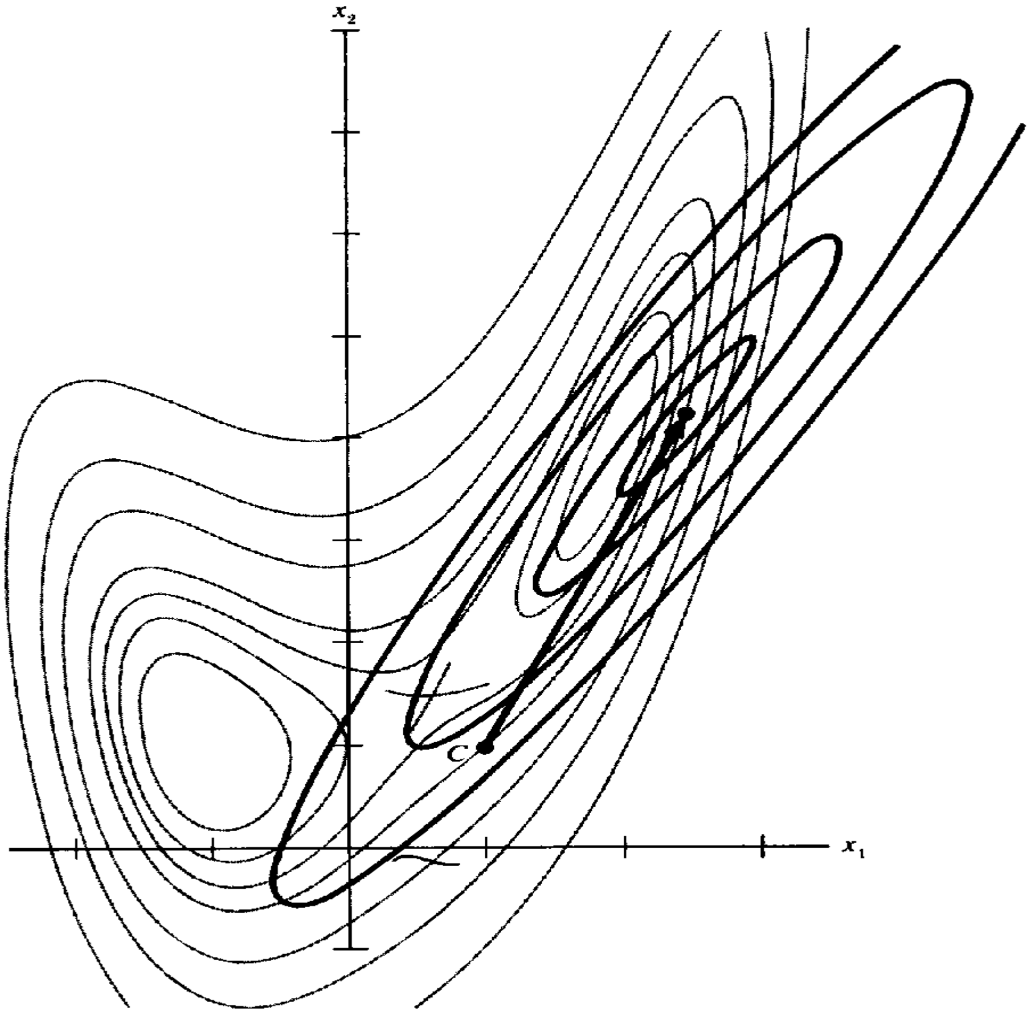
Tehnike numerične optimizacije

Primer 2. Newton, trajektorija iskanja iz prve začetne točke. Druga iteracija, optimizacija se ustavi v sedlu.



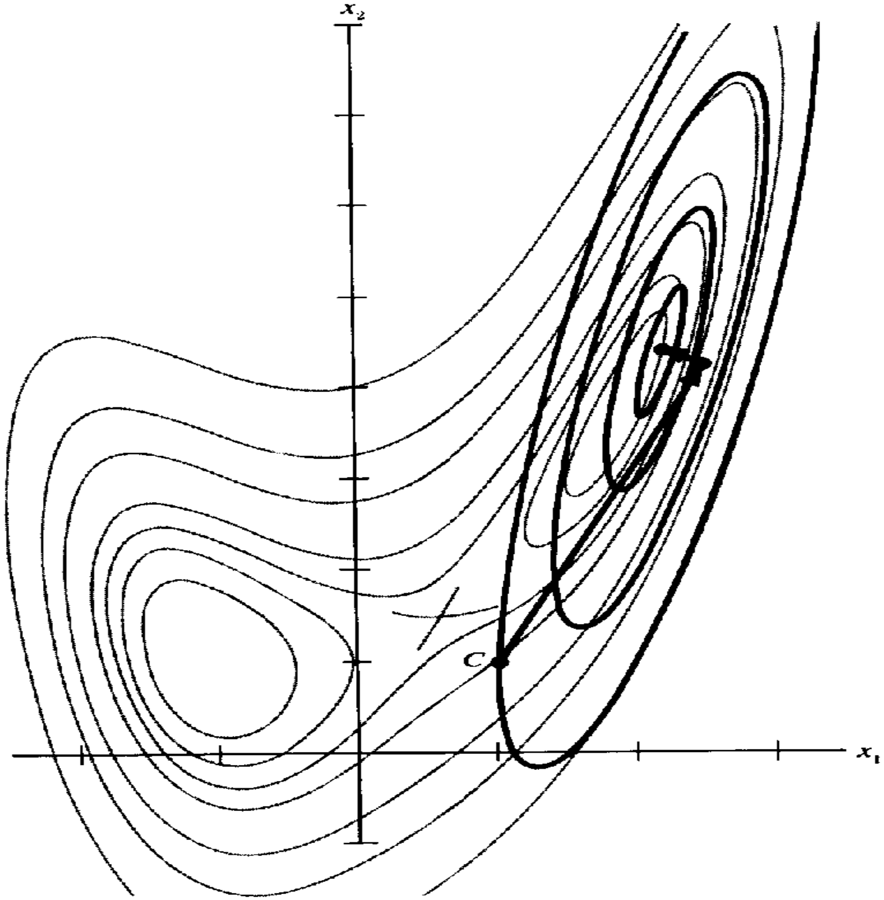
Tehnike numerične optimizacije

Primer 2. Newton, trajektorija iskanja iz tretje začetne točke. Prva iteracija.



Tehnike numerične optimizacije

Primer 2. Newton, trajektorija iskanja iz tretje začetne točke. Druga iteracija.



Tehnike numerične optimizacije

Sankcijska funkcija se približuje zunanji sankcijski funkciji

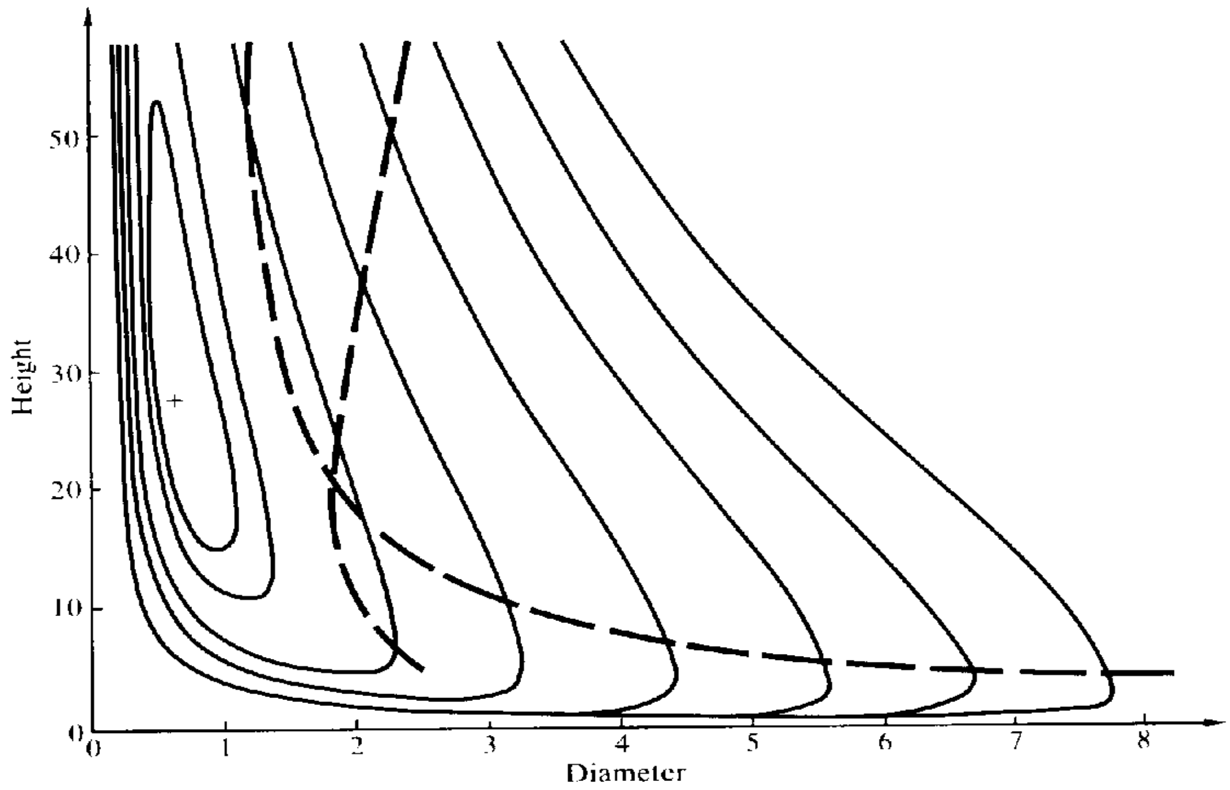
$$\Phi_i(\boldsymbol{x}) = F(\boldsymbol{x}) + r_i P(\boldsymbol{x})$$

$$P(\boldsymbol{x}) = \sum_{j=1}^{M} \langle G_i(\boldsymbol{x}) - 1 \rangle^2$$

$$\langle A(\boldsymbol{x}) \rangle = \begin{cases} A(\boldsymbol{x}) \text{ if } & A(\boldsymbol{x}) > 0 \\ 0 \text{ if } & A(\boldsymbol{x}) \leq 0 \end{cases}$$

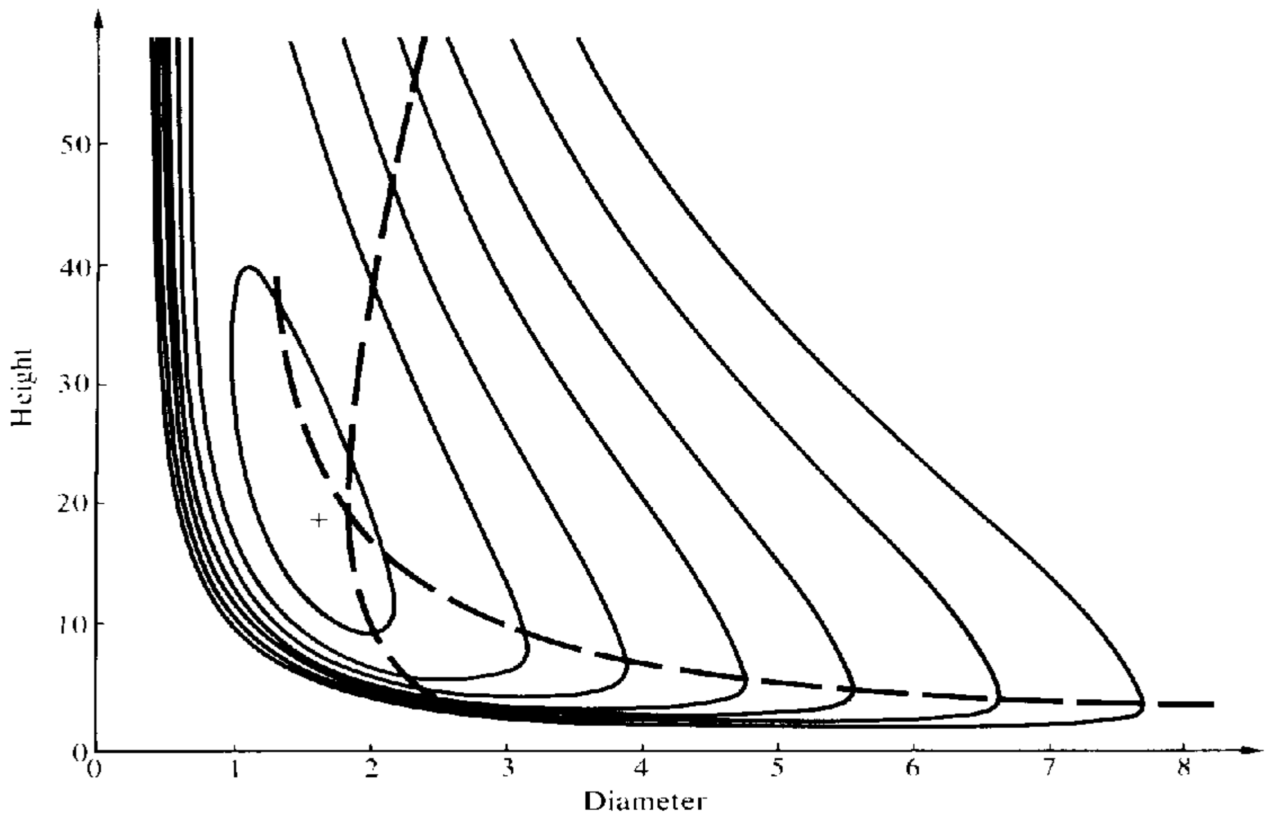
Tehnike numerične optimizacije

Približevanje zunanji sankcijski funkciji. Prva iteracija.



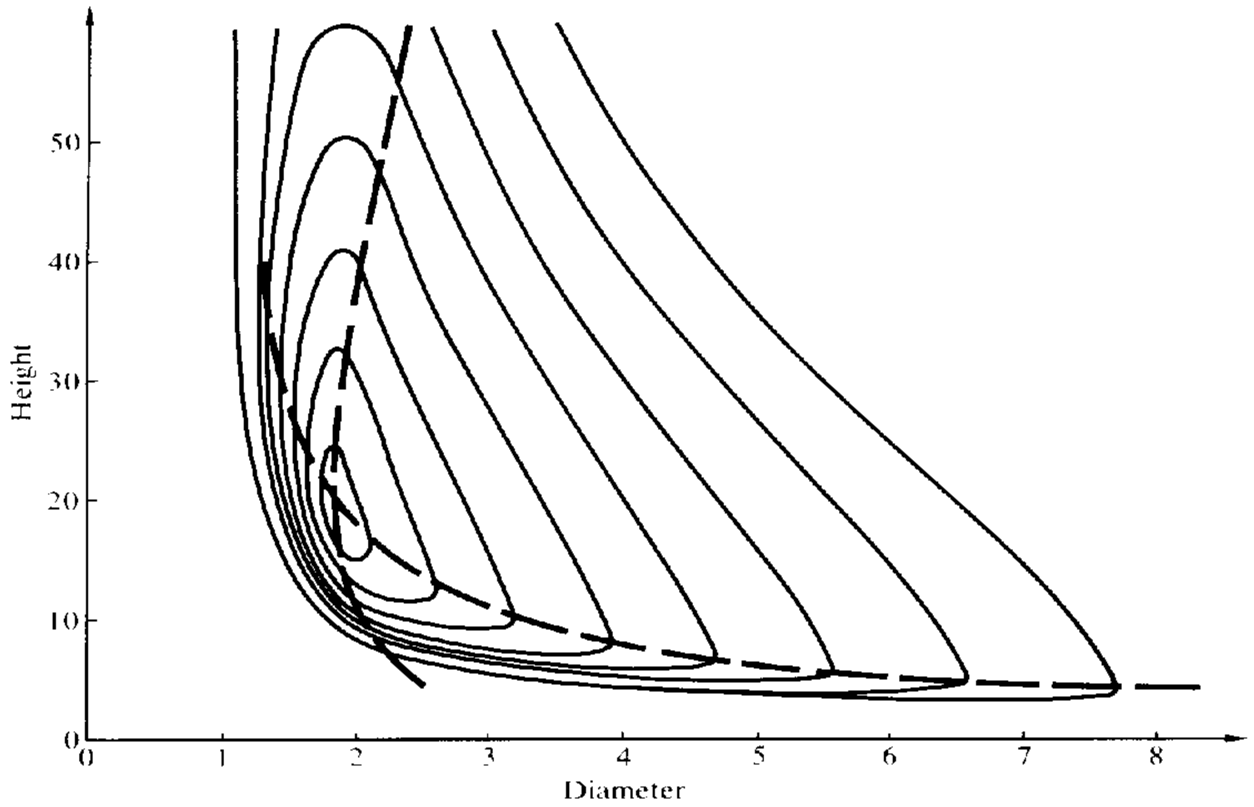
Tehnike numerične optimizacije

Približevanje zunanji sankcijski funkciji. Druga iteracija.



Tehnike numerične optimizacije

Približevanje zunanji sankcijski funkciji. Tretja iteracija.



Tehnike numerične optimizacije

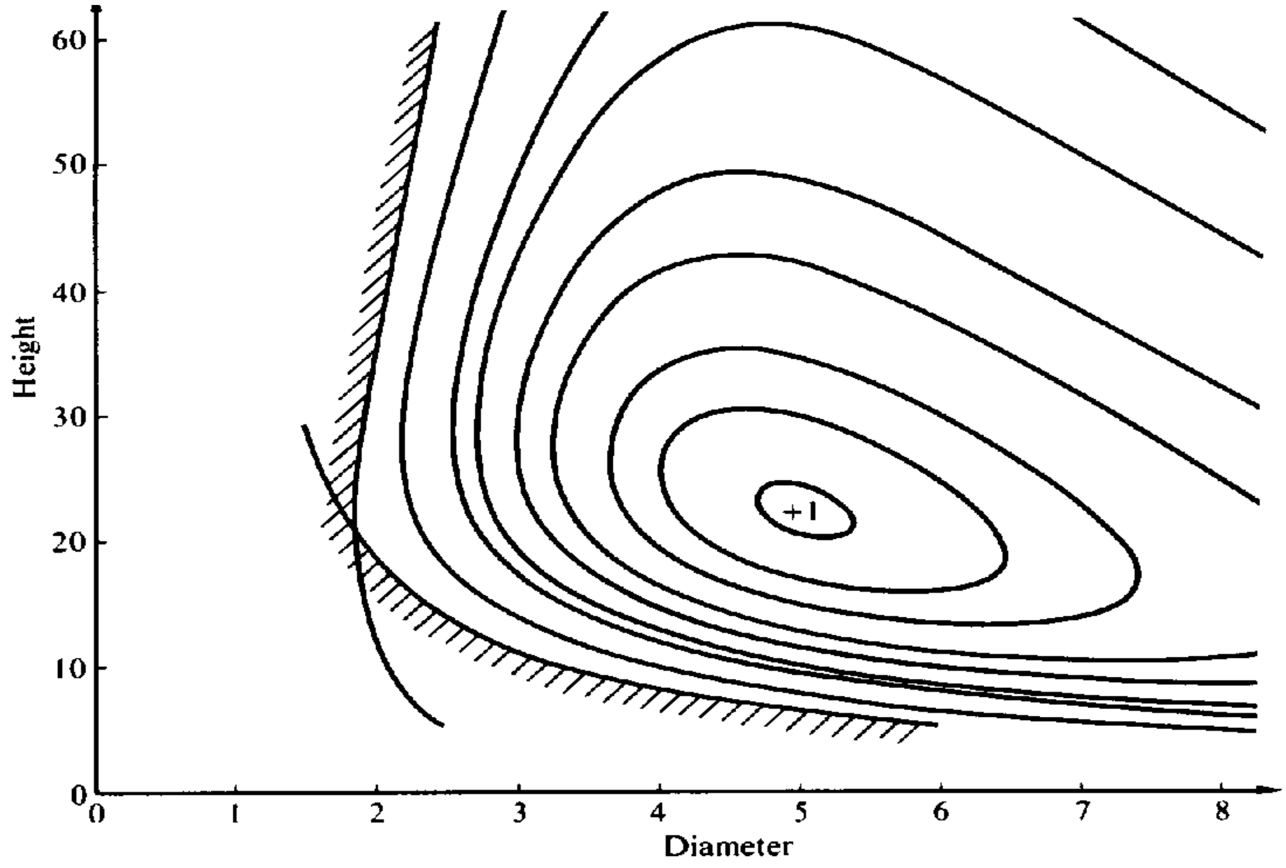
Sankcijska funkcija se približuje notranji sankcijski funkciji

$$\Phi_i(\boldsymbol{x}) = F(\boldsymbol{x}) + r_i P(\boldsymbol{x})$$

$$P(\boldsymbol{x}) = \sum_{j=1}^{M} \frac{1}{\left(G_i(\boldsymbol{x}) - 1\right)^2}$$

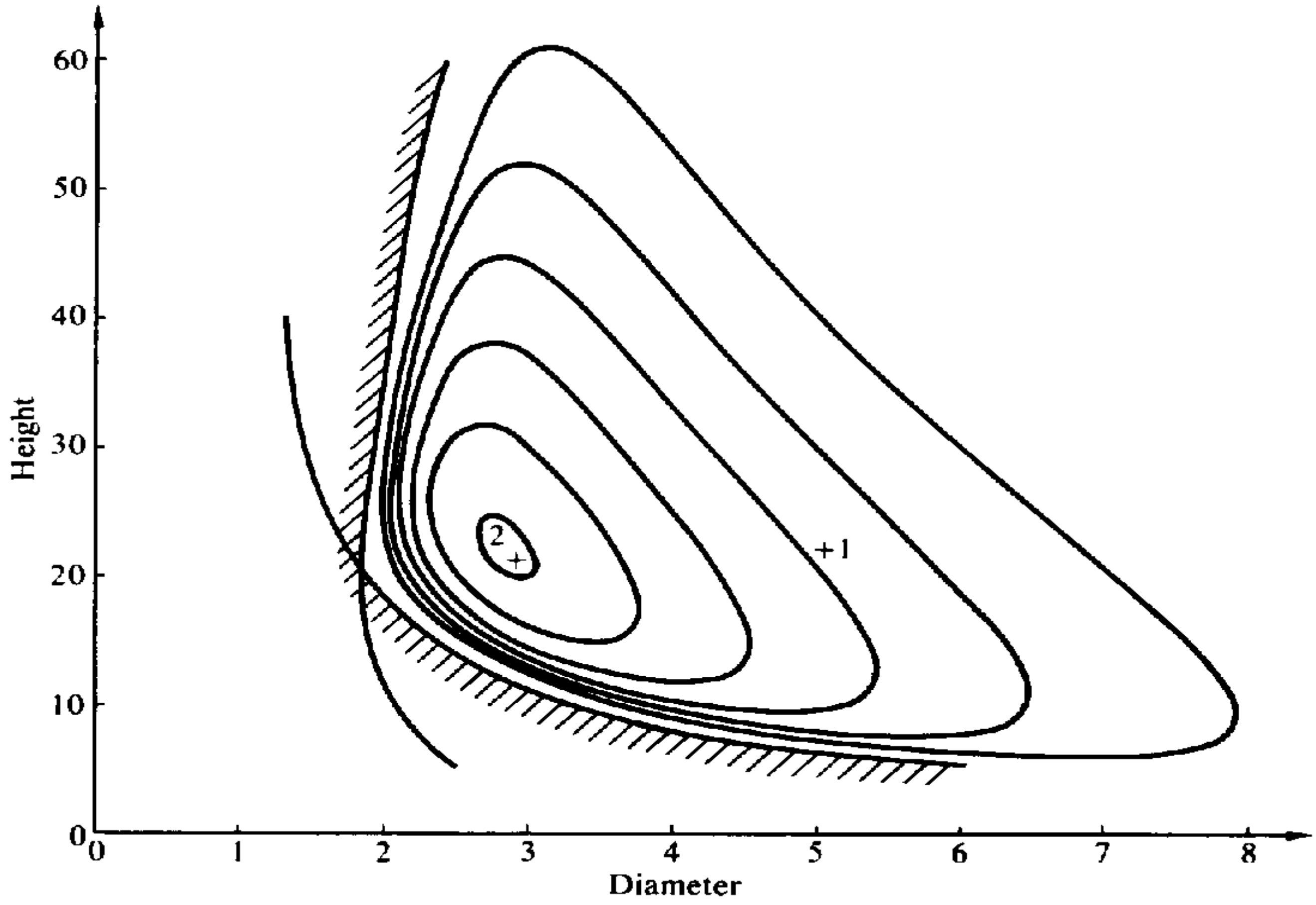
Tehnike numerične optimizacije

Približevanje notranji sankcijski funkciji. Prva iteracija.



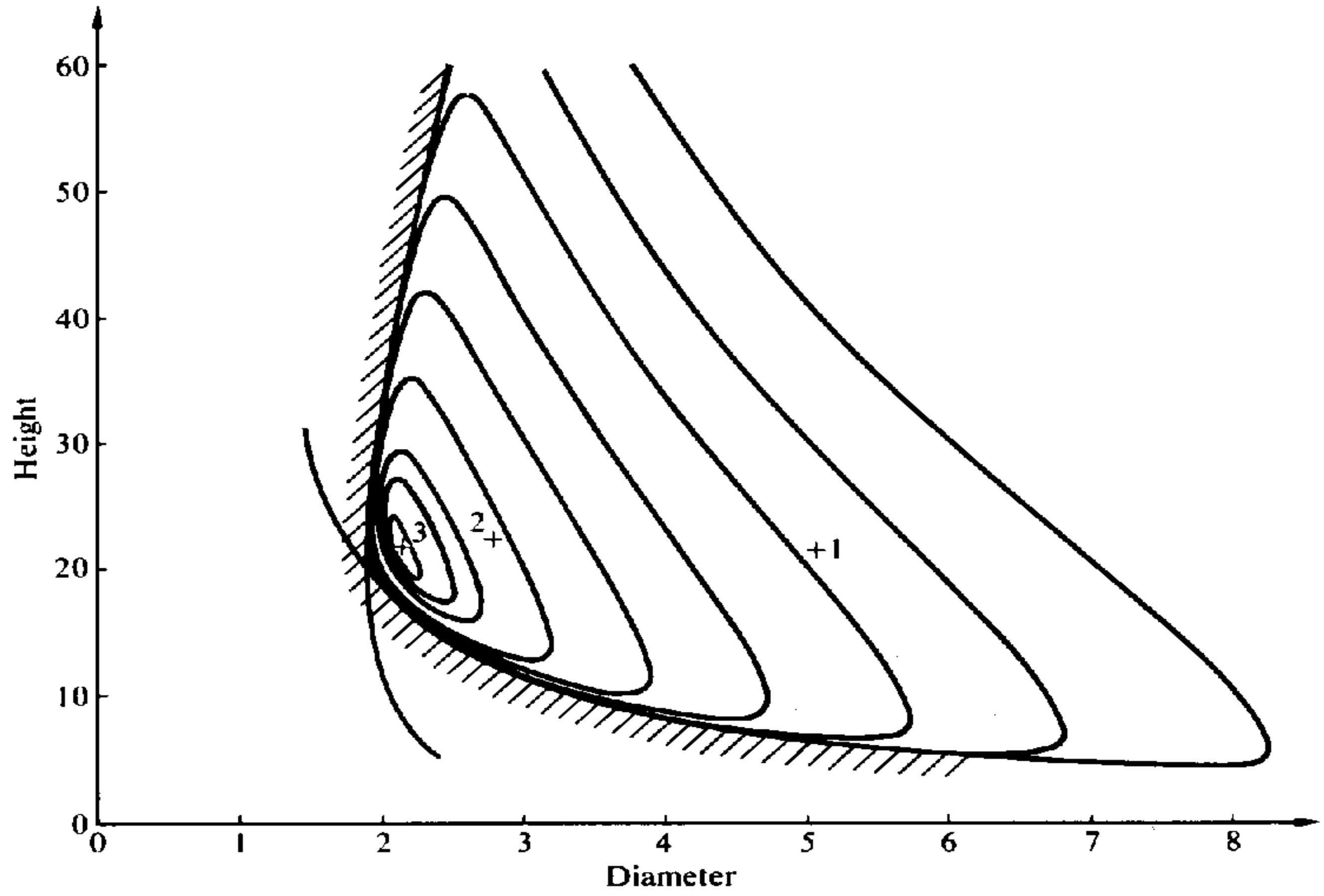
Tehnike numerične optimizacije

Približevanje notranji sankcijski funkciji. Druga iteracija.



Tehnike numerične optimizacije

Približevanje notranji sankcijski funkciji. Tretja iteracija.



•

Vprašanja?