

## Optymalizacja (minimalizacja) funkcji

Plan wykładu:

1. Sformułowanie problemu, funkcja celu
2. Metody bezgradientowe
  - a) metoda złotego podziału
  - b) metoda sympleks
  - c) metoda interpolacji Powell'a
3. Metody gradientowe
  - a) metoda największego spadku
  - b) metoda gradientów sprzężonych
  - c) metoda Newtona
4. Minimalizacja z ograniczeniami:  
metoda funkcji kary zewnętrznej i wewnętrznej.
5. Metody stochastyczne (symulowane wyżarzanie, algorytmy genetyczne)  
-> metody Monte Carlo

## Sformułowanie problemu, definicje pomocnicze

Zadaniem **optymalizacji** jest poszukiwanie minimum lub maksimum funkcji (wielu zmiennych).

W praktyce problem sprowadza się do poszukiwania minimum czyli takiego punktu dla którego zachodzi

$$f : R^n \rightarrow R$$

$$\min f(\mathbf{x}) = f(\mathbf{x}^*) \Leftrightarrow \bigwedge_{\mathbf{x} \in R^n} f(\mathbf{x}^*) < f(\mathbf{x})$$

$$\mathbf{x} = [x_1, x_2, \dots, x_n]^T$$

z warunkami

$$g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, m$$

$$h_j(\mathbf{x}) = 0, \quad j = 1, 2, \dots, r$$

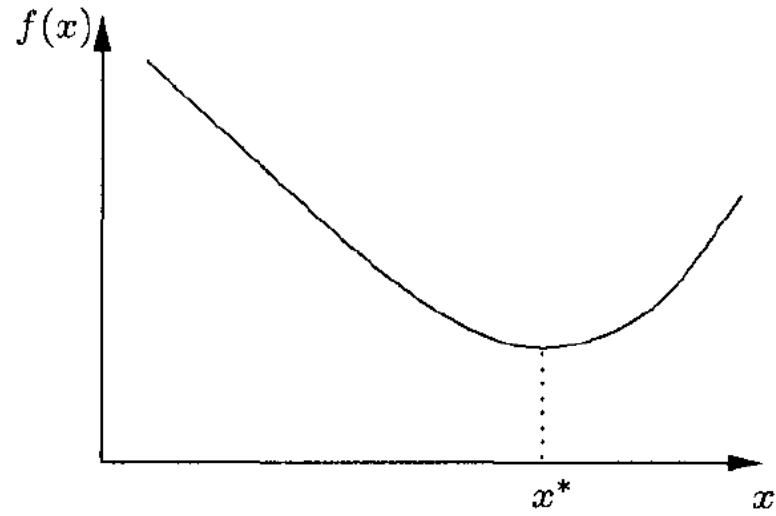
Funkcje:  $f(\mathbf{x}), g(\mathbf{x}), h(\mathbf{x})$  są funkcjami skalarnymi.

$f(\mathbf{x})$  – funkcja celu, celem jest znalezienie jej minimum (optymalizacja)

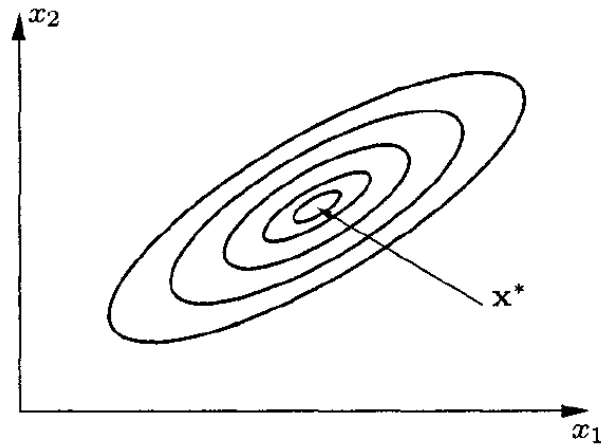
$g(\mathbf{x})$  i  $h(\mathbf{x})$  - funkcje określające warunki jakie musi spełniać rozwiązanie (**więzy**) – ograniczają przestrzeń dopuszczalnych rozwiązań

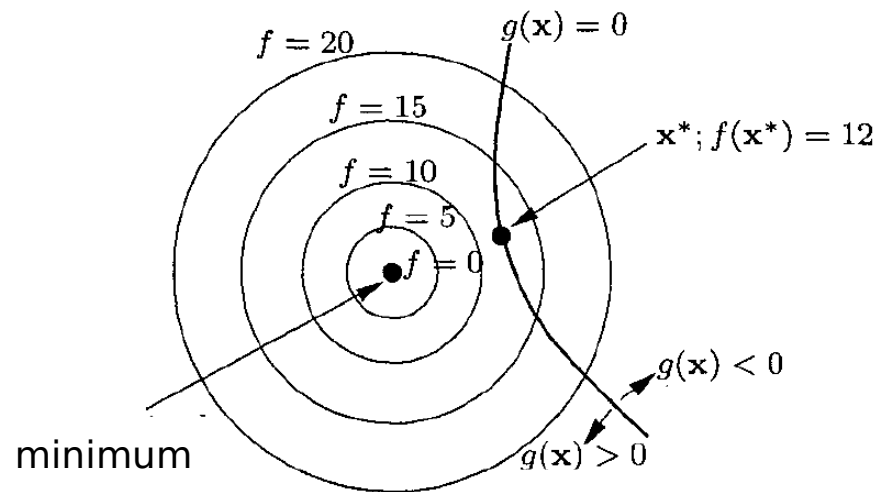
## Przykład

Problem jednowymiarowy



Problem dwuwymiarowy





**Rys.** Przykład poszukiwania minimum z nałożonymi warunkami na rozwiązanie

Trzy przypadki:

- 1) Problem bez więzów oraz dla  $g(x) > 0$   
minimum znajdujemy dla  $f(x) = 0$
- 2) Jeśli warunkiem jest  $g(x) = 0$  to minimum  
znajduje się w punkcie takim że  $f(x) = 12$
- 3) Jeśli warunkiem jest  $g(x) < 0$  to rozwiązanie znajdziemy  
w pobliżu punktu w którym  $f(x) = 12$

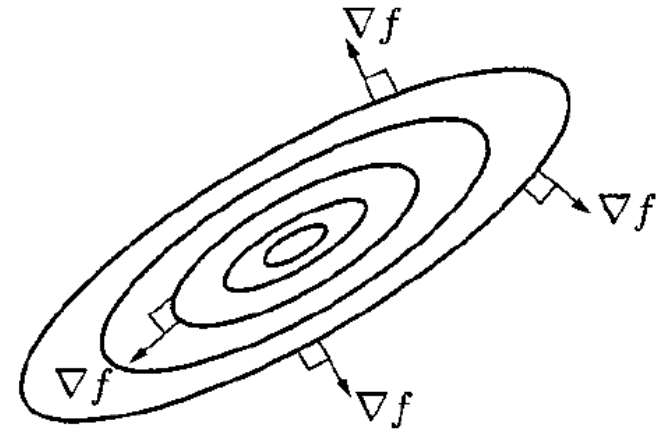
## Gradient funkcji - wektor gradientu

Dla funkcji celu

$$f(\mathbf{x}) \in C^2$$

definiujemy funkcję wektorową będącą gradientem funkcji

$$\mathbf{g}(\mathbf{x}) = \nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \frac{\partial f(\mathbf{x})}{\partial x_2} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_n} \end{bmatrix}$$



Uwaga:  
gradient skierowany jest zawsze  
w stronę narastających wartości

## Hesjan (macierz Hessego)

Dla funkcji celu

$$f(\mathbf{x}) \in C^2$$

definiujemy macierz (**hesjan**) której elementami są jej drugie pochodne cząstkowe

$$\begin{aligned} H(\mathbf{x}) &= \left\{ \frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j} \right\} = \nabla \nabla^T f(\mathbf{x}) \\ &= \begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} & \cdots & \cdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_1} & \cdots & \cdots & \cdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_1} & \cdots & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n^2} \end{bmatrix} \end{aligned}$$

Macierz  $H(\mathbf{x})$  jest symetryczna – implikacje numeryczne.

## Minimum lokalne, minimum globalne oraz punkt siodłowy funkcji celu

a) punkt  $\mathbf{x}^*$  stanowi **minimum globalne** funkcji jeśli

$$\bigwedge_{\mathbf{x} \in R^n} f(\mathbf{x}) \geq f(\mathbf{x}^*)$$

b) punkt  $\mathbf{x}^*$  stanowi **minimum lokalne** funkcji jeśli

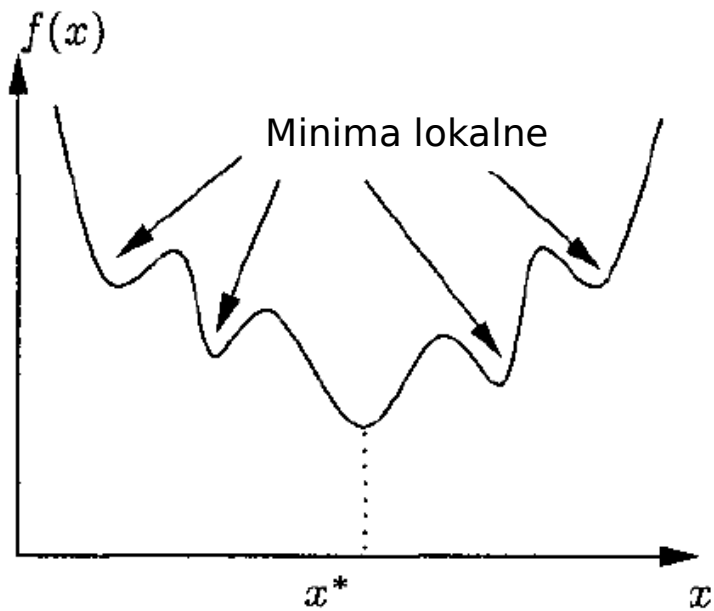
$$\exists \varepsilon : \varepsilon > 0, \varepsilon \in R \quad \bigwedge_{\mathbf{x} : \|\mathbf{x} - \mathbf{x}^*\| < \varepsilon} f(\mathbf{x}) > f(\mathbf{x}^*)$$

c) punkt

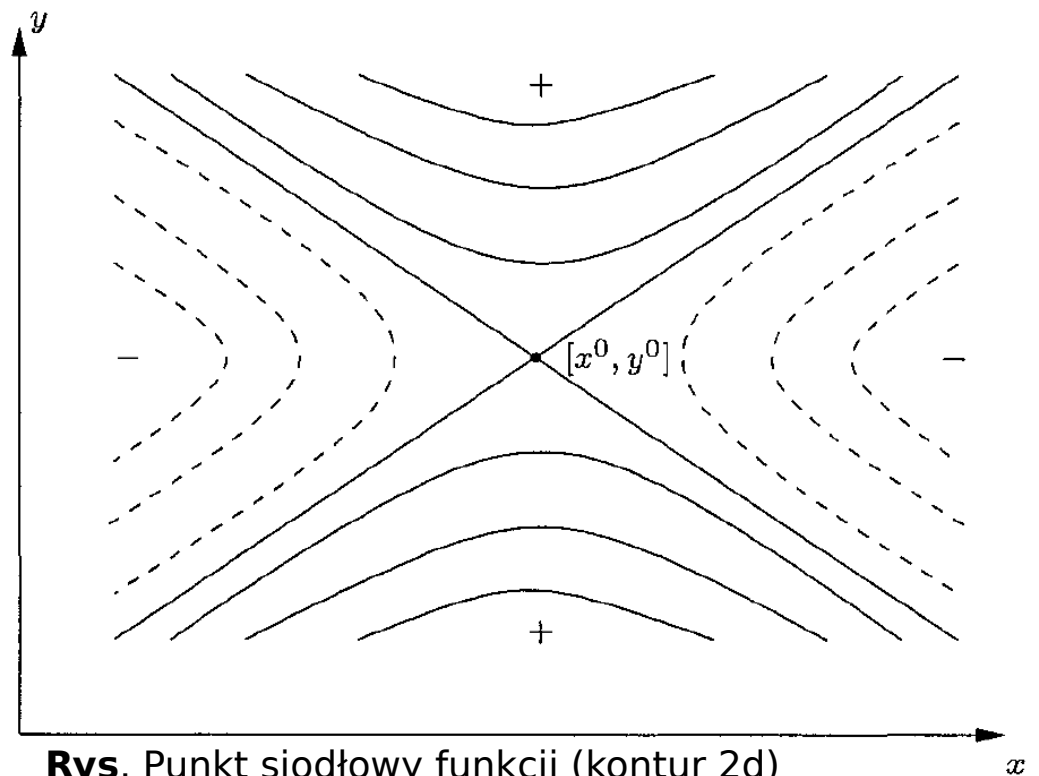
$$\mathbf{x}^* = \begin{pmatrix} x^0 \\ y^0 \end{pmatrix}$$

jest **punktem siodłowym** funkcji jeśli

$$\exists \varepsilon : \varepsilon > 0, \varepsilon \in R \quad \bigwedge_{\substack{\mathbf{x} : \|\mathbf{x} - \mathbf{x}^0\| < \varepsilon \\ \mathbf{y} : \|\mathbf{y} - \mathbf{y}^0\| < \varepsilon}} f(\mathbf{x}, \mathbf{y}^0) \leq f(\mathbf{x}^0, \mathbf{y}^0) \leq f(\mathbf{x}^0, \mathbf{y})$$



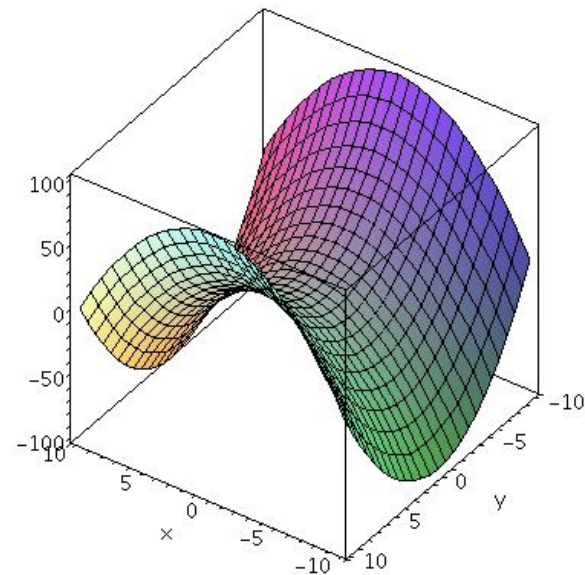
**Rys.** Minima lokalne i minimum globalne



**Rys.** Punkt siodłowy funkcji (kontur 2d)

**Rys.** Punkt siodłowy w 3D

$$f(x, y) = y^2 - x^2$$





## Metoda złotego podziału (metoda jednowymiarowa, niegradientowa)

- 1) Wstępnie wyznaczamy przedział  $[a,b]$  w którym spodziewamy się minimum wartości funkcji
- 2) W przedziale  $[a,b]$  wyznaczamy dwa punkty  $\lambda_1$  i  $\lambda_2$

3) Jeśli

$$F(\lambda_2) > F(\lambda_1)$$

to zmieniamy granice przedziału na  **$[a, \lambda_2]$**

4) Jeśli

$$F(\lambda_2) < F(\lambda_1)$$

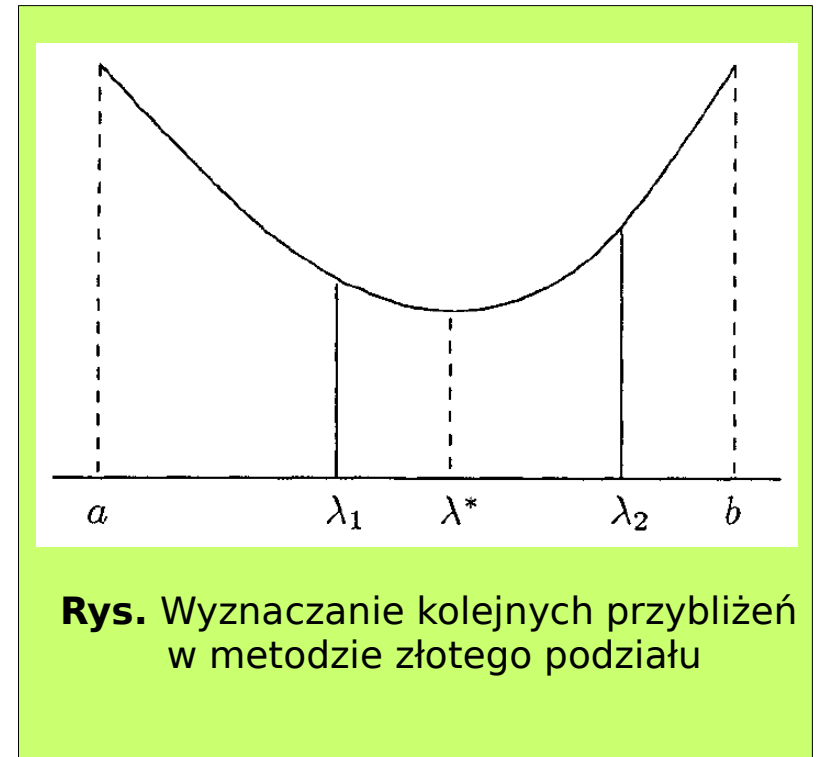
to zmieniamy granice przedziału na  **$[\lambda_1, b]$**

- 5) Proces podziału prowadzimy iteracyjnie aż do spełnienia warunku

$$|a^i - b^i| < \varepsilon$$

a jako przybliżenie minimum możemy przyjąć

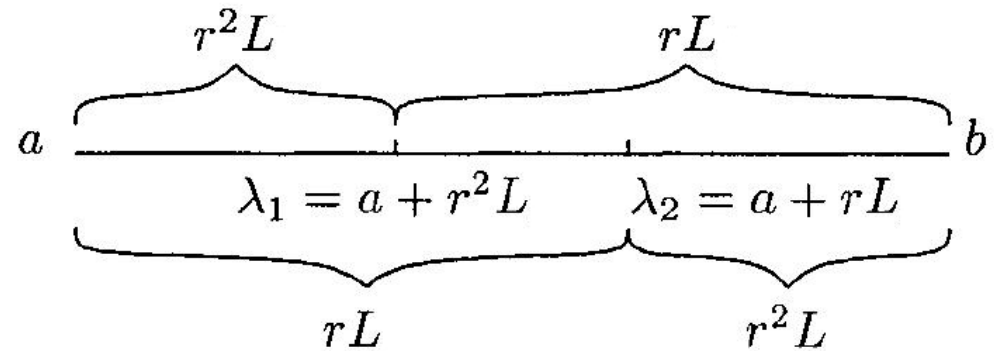
$$\lambda^* = \frac{a^i + b^i}{2}$$



Pozostaje tylko kwestia jak wyznaczyć punkty tak aby wybór był optymalny tzn. **chcemy wykonać jak najmniejszą ilość podziałów.**

Punktem wyjścia jest zależność  
(złota proporcja/podział)

$$\frac{(\lambda_1 - a) + (b - \lambda_1)}{b - \lambda_1} = \frac{b - \lambda_1}{\lambda_1 - a} = \varphi$$



Uzależnimy b od a

$$b - a = L \Rightarrow b = L + a$$

po wstawieniu do równania otrzymujemy

$$\frac{L}{L + a - \lambda_1} = \frac{L + a - \lambda_1}{\lambda_1 - a}$$

$$L(\lambda_1 - a) = (L - (\lambda_1 - a))^2$$

$$(\lambda_1 - a) = L \underbrace{\left(1 - \frac{(\lambda_1 - a)}{L}\right)^2}_{=r^2} = Lr^2$$

Na podstawie rysunku możemy zapisać drugą relację

$$b - \lambda_1 = L - (\lambda_1 - a)$$

$$b - \lambda_1 = L \left(1 - \frac{(\lambda_1 - a)}{L}\right) = Lr$$

Otrzymaliśmy dwie zależności

$$(\lambda_1 - a) = Lr^2$$

$$(b - \lambda_1) = Lr$$

po wstawieniu ich do równania wyjściowego dostajemy równanie kwadratowe na „r”

$$\frac{Lr^2 + Lr}{Lr} = \frac{Lr}{Lr^2} = \frac{1}{r} \quad \Rightarrow \quad r^2 + r - 1 = 0$$

i znajdujemy jego pierwiastki

$$r_1 = \frac{\sqrt{5} - 1}{2} = 0.618034 > 0$$

(dodatni - zachowujemy)

$$r_2 = \frac{-\sqrt{5} - 1}{2} < 0$$

(ujemny - odrzucamy)

Po wyborze  $r=r_1$  możemy określić wartości  $\lambda_1$  i  $\lambda_2$  zakładając ponadto, że oba punkty powinny być symetryczne względem krańców przedziału

$$\lambda_1 = a + r^2L$$

$$\lambda_2 = a + rL$$

## Metoda interpolacji kwadratowej Powell'a

Przez trzy punkty:  $\lambda_1, \lambda_2, \lambda_3$  prowadzimy wielomian 2 stopnia

$$p_2(\lambda) = F(\lambda_0) + F[\lambda_0, \lambda_1](\lambda - \lambda_0) + F[\lambda_0, \lambda_1, \lambda_2](\lambda - \lambda_0)(\lambda - \lambda_1)$$

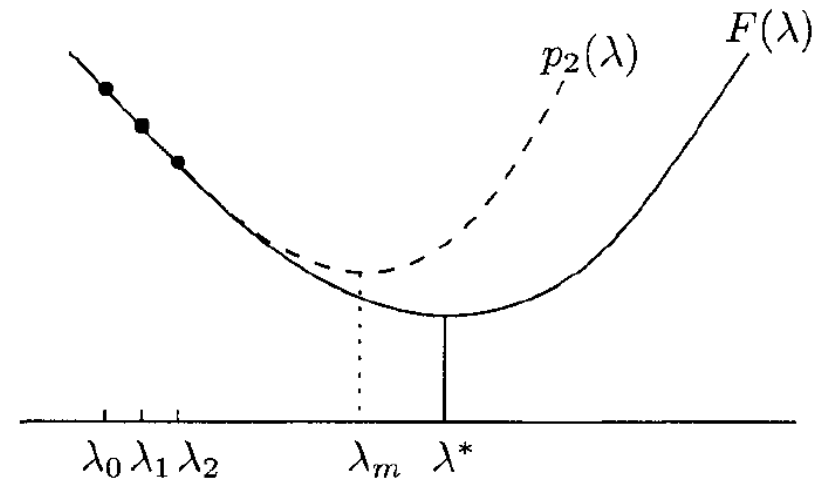
gdzie:  $F(\lambda_0)$  - wartość funkcji  
 $F[\lambda_0, \lambda_1]$  - iloraz różnicowy 1 rzędu,  
 $F[\lambda_0, \lambda_1, \lambda_2]$  - iloraz różnicowy 2 rzędu

Narzucamy warunek zerowania się pochodnej (spodziewamy się minimum)

$$\frac{dp_2}{d\lambda} = F[\lambda_0, \lambda_1] + 2\lambda F[\lambda_0, \lambda_1, \lambda_2] - F[\lambda_0, \lambda_1, \lambda_2](\lambda_0 + \lambda_1) = 0$$

rozwiązując to równanie ze względu na  $\lambda$  otrzymamy

$$\lambda_m = \frac{F[\lambda_0, \lambda_1, \lambda_2](\lambda_0 + \lambda_1) - F[\lambda_0, \lambda_1]}{2F[\lambda_0, \lambda_1, \lambda_2]} \approx \lambda^*$$



**Rys.** Wyznaczanie przybliżonego rozwiązania w metodzie Powell'a.

Aby znaleziony punkt był rzeczywistym minimum, iloraz  $(F[\lambda_0, \lambda_1, \lambda_2])$  musi spełniać warunek

$$F[\lambda_0, \lambda_1, \lambda_2] > 0$$

## Algorytm interpolacji Powell'a:

1) Wybierz  $\lambda_0$  i oblicz

$$F[\lambda_0+h] < F[\lambda_0], \quad F[\lambda_0+2h] < F[\lambda_0+h]$$

(ewentualnie zmień znak:  $-h$ , jeśli nierówności nie są spełnione)

2) Wyznacz  $\lambda_m$  i sprawdź czy jest minimum

3) Jeśli

$$|\lambda_m - \lambda_n| > h$$

odrzuć najdalej położony od  $\lambda_m$  punkt i ponownie wykonaj obliczenia z pkt. 2.  
 $\lambda_n$  - najbliższy położony punkt względem  $\lambda_m$

Punkt  $\lambda_m$  akceptujemy jako minimum jeśli

$$|\lambda_m - \lambda_n| < \varepsilon$$

## Metoda sympleks (Nelder-Mead)

Rozwiązania poszukujemy iteracyjnie dążąc do znalezienia minimum funkcji poprzez obliczanie jej wartości w ściśle określonych punktach. Do wyznaczenia kierunków poszukiwań oraz punktów, w których oblicza się wartości funkcji wykorzystuje się obiekt zwany **sympleksem**.

**Sympleks** jest to  $n$ -wymiarowy obiekt, stanowiący najmniejszy zbiór wypukły zawierający punkty

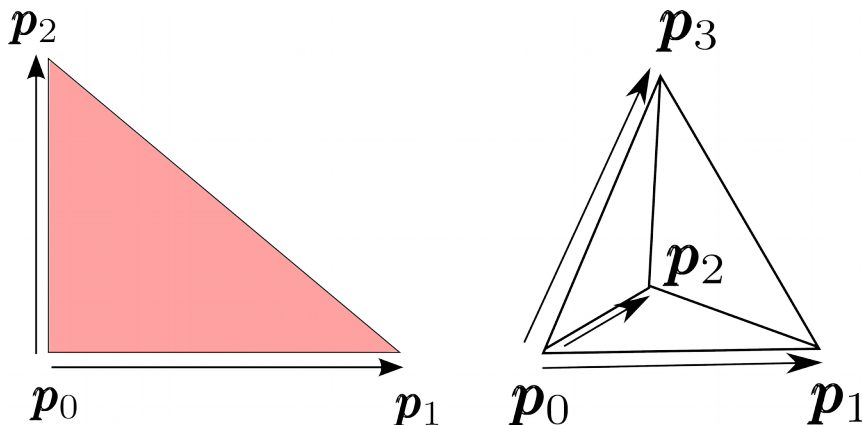
$$p_0, p_1, \dots, p_n$$

będące jego wierzchołkami. Wierzchołki sympleksu tworzą tylko takie punkty, dla których zbiór wektorów

$$\{p_i - p_0\}$$

jest liniowo niezależny (punkty nie są współliniowe).

**Przykład.** Sympleksy w 2 i w 3 wymiarach



Sposób postępowania w jednej iteracji

1. wybieramy zestaw  $N+1$  punktów startowych ( $N$  - liczba wymiarów), np. w postaci

$$\mathbf{p}_i = \mathbf{p}_0 + \lambda_i \mathbf{d}_i, \quad i = 1, 2, \dots, n$$

wektory  $\mathbf{d}_i$  możemy przyjąć jako wersory układu kartezjańskiego

$$\mathbf{d}_i = \mathbf{e}_i, \quad i = 1, 2, \dots, n$$

Położenie  $\mathbf{p}_0$  wybieramy arbitralnie, wartości  $\lambda_i$  zależą od rozpatrywanego problemu.

2. Dla wszystkich wierzchołków obliczamy ciąg

$$f(\mathbf{p}_0), f(\mathbf{p}_1), \dots, f(\mathbf{p}_N)$$

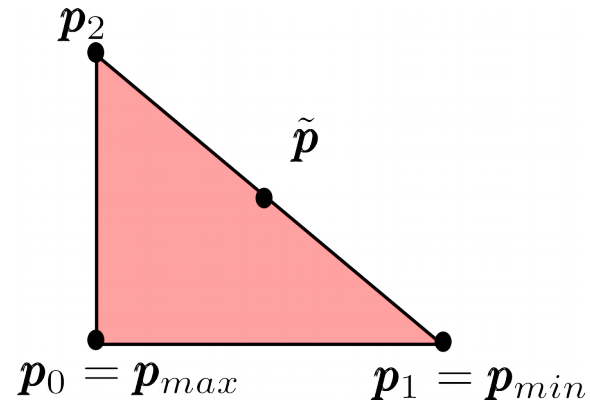
i szukamy dwóch punktów odpowiednio dla największej i najmniejszej wartości funkcji

$$\mathbf{p}_{min} = \mathbf{p}_j \iff f(\mathbf{p}_j) \leq f(\mathbf{p}_i), \quad i = 0, 1, \dots, N$$

$$\mathbf{p}_{max} = \mathbf{p}_j \iff f(\mathbf{p}_j) \geq f(\mathbf{p}_i), \quad i = 0, 1, \dots, N$$

3. Wyznaczamy środek ciężkości wierzchołków z wyłączeniem punktu  $\mathbf{p}_{max}$

$$\tilde{\mathbf{p}} = \frac{1}{n} \sum_{i \neq max} \mathbf{p}_i$$



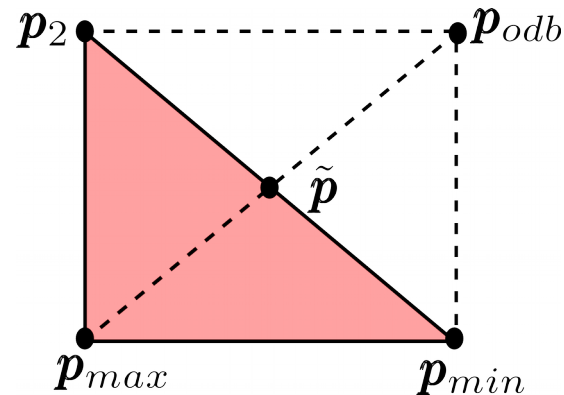
## 4. Odbicie

Wyznaczamy punkt  $\mathbf{p}_{odb}$  leżący na linii łączącej środek ciężkości i wierzchołek  $\mathbf{p}_{max}$

$$\mathbf{p}_{odb} = \tilde{\mathbf{p}} + \alpha(\tilde{\mathbf{p}} - \mathbf{p}_{max}), \quad \alpha \in [0, 1)$$

i jeśli zachodzi warunek

$$f(\mathbf{p}_{min}) \leq f(\mathbf{p}_{odb}) < f(\mathbf{p}_{max})$$



to odbicie akceptujemy. W nowym sympleksie wierzchołek  $\mathbf{p}_{max}$  zostaje zastąpiony przez  $\mathbf{p}_{odb}$ .

## 5. Ekspansja

Po odbiciu sprawdzamy też czy zachodzi warunek

$$f(\mathbf{p}_{odb}) < f(\mathbf{p}_{min})$$

Jeśli tak to kierunek poszukiwań jest dobry i wtedy zamiast  $\mathbf{p}_{odb}$  wyznaczamy

$$\mathbf{e} = \tilde{\mathbf{p}} + \gamma(\mathbf{p}_{odb} - \tilde{\mathbf{p}}), \quad \gamma > 1$$



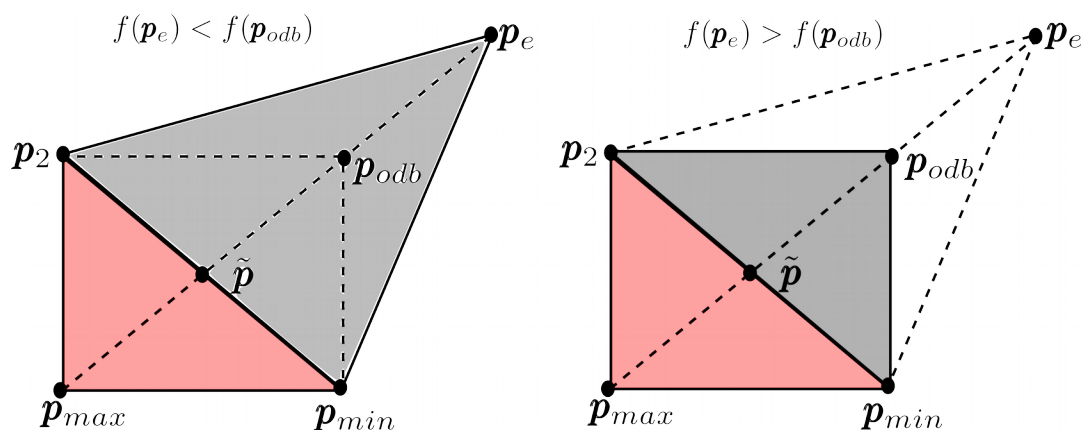
Jeśli

$$f(\mathbf{p}_e) < f(\mathbf{p}_{odb})$$

to wierzchołki nowego sympleksu tworzą poprzednie za wyjątkiem wierzchołka  $\mathbf{p}_{max}$  który zastępujemy  $\mathbf{p}_e$ . W przeciwnym wypadku

$$f(\mathbf{p}_e) \geq f(\mathbf{p}_{odb})$$

wierzchołek  $\mathbf{p}_{max}$  zastępujemy  $\mathbf{p}_{odb}$



## 6. Zawężenie

Jeśli

$$f(\mathbf{p}_{odb}) \geq f(\mathbf{p}_{max})$$

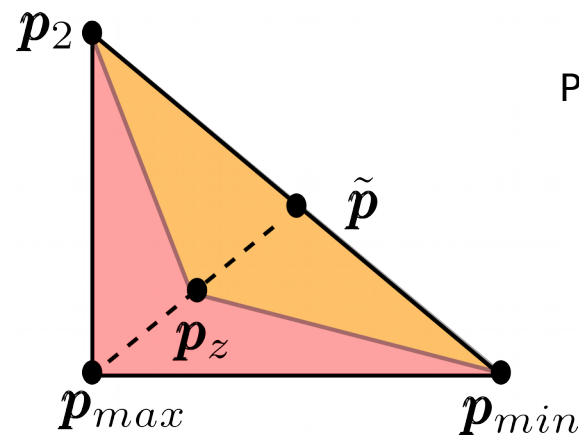
to należy wykonać tzw. zawężenie sympleksu. Wyznaczamy nowe przybliżenie

$$\mathbf{p}_{odb} = \tilde{\mathbf{p}} + \beta(\mathbf{p}_{max} - \tilde{\mathbf{p}}), \beta \in (0, 1)$$

Jeśli zachodzi warunek

$$f(\mathbf{p}_z) < f(\mathbf{p}_{max})$$

to wierzchołek  $\mathbf{p}_{max}$  zastępujemy  $\mathbf{p}_z$ .



Przykład zawężenia.

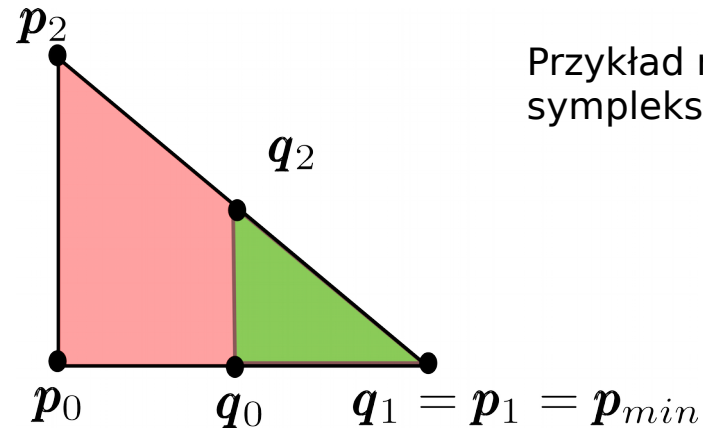
## 7. Redukcja

Jeśli

$$f(\mathbf{p}_z) \geq f(\mathbf{p}_{max})$$

dokonyjemy redukcji sympleksu

$$\begin{aligned} \mathbf{p}_i &\leftarrow \delta(\mathbf{p}_i + \mathbf{p}_{min}), & i = 0, 1, \dots, n, \\ & & i \neq min \\ & & \delta \in (0, 1) \end{aligned}$$



Przykład redukcji sympleksu.

Przy sprawdzaniu warunków akceptacji nowego wierzchołka sympleksu sprawdzamy też warunek stopu jako kryterium stopu przyjmując

$$\max_{i=0,1,\dots,n} \|\mathbf{p}_{min} - \mathbf{p}_i\|_2 < \varepsilon$$

**Metoda sympleks jest mało efektywna, nierzadko potrzeba dużej liczby iteracji w celu znalezienia minimum funkcji - nie wymaga jednak liczenia pochodnych.**

## Kierunkowe metody poszukiwania minimum funkcji

### Pochodna kierunkowa funkcji celu

Różniczkę zupełną funkcji celu (skalar) możemy zdefiniować jako iloczyn skalarny wektorów

$$df = \frac{\partial f}{\partial x_1} dx_1 + \dots + \frac{\partial f}{\partial x_n} dx_n = \nabla f(\mathbf{x}) d\mathbf{x}$$

Jeśli wektor  $\mathbf{u}$  wyznacza kierunek prostej łączącej punkty  $\mathbf{x}$  i  $\mathbf{x}'$  to są one ze sobą powiązane

$$\mathbf{x}(\lambda) = \mathbf{x}' + \lambda \mathbf{u}$$

Natomiast dla bardzo małych zmian wartości  $\lambda$  możemy zapisać

$$d\mathbf{x} = \mathbf{u} d\lambda$$

Na prostej łączącej (**ustalone punkty**)  $\mathbf{x}$  i  $\mathbf{x}'$  wartość funkcji celu zależna będzie od nowej zmiennej ( $\lambda$ )

$$F(\lambda) = f(\mathbf{x}' + \lambda \mathbf{u}) = f(\mathbf{x})$$

Liczmy różniczkę zupełną dla funkcji celu zależnej od  $\lambda$

$$dF = df = \nabla^T f(\mathbf{x}) \mathbf{u} d\lambda$$

Możemy teraz wyrazić pochodną kierunkową funkcji celu w punkcie  $\mathbf{x}$  dla kierunku  $\mathbf{u}$  następująco

$$\frac{dF(\lambda)}{d\lambda} = \left. \frac{df(\mathbf{x})}{d\lambda} \right|_{\mathbf{u}} = \nabla^T f(\mathbf{x})\mathbf{u}$$

Skorzystajmy z rozwinięcia w szereg Taylora

$$F(\lambda) = F(0) + \frac{dF}{d\lambda} \lambda + \underbrace{\frac{d^2 F}{d\lambda^2} \frac{\lambda^2}{2} + \dots}_{\text{zaniedbujemy}}$$

$$F(\lambda) = f(\mathbf{x}' + \lambda\mathbf{u}) = f(\mathbf{x})$$

$$F(0) = f(\mathbf{x}' + \mathbf{0} \cdot \mathbf{u}) = f(\mathbf{x}')$$

i załóżmy

$$\lambda > 0, \quad \frac{dF}{d\lambda} = \nabla^T f(\mathbf{x}')\mathbf{u} = - \left| \nabla^T f(\mathbf{x}')\mathbf{u} \right| < 0$$

wówczas otrzymamy

$$F(\lambda) = f(\mathbf{x}') - \lambda \left| \nabla^T f(\mathbf{x}')\mathbf{u} \right| < f(\mathbf{x}')$$

**Wniosek: ponieważ gradient funkcji jest określony w dowolnym punkcie, należy tak dobrać wektor  $\mathbf{u}$  aby spełnić warunek**

$$\nabla^T f(\mathbf{x}')\mathbf{u} < 0$$

Przykład:  $\mathbf{u} = -\nabla f(\mathbf{x}') \implies -\|\nabla f(\mathbf{x}')\| < 0$

taki warunek  
odpowiada metodzie  
największego spadku  
(**Steepest Descent**)

## Jak wykorzystać pochodną kierunkową do znalezienia minimum funkcji?

Startując z określonego punktu  $\mathbf{x}_0$  poszukujemy kolejnego przybliżenia tj.  $\mathbf{x}_1$  w kierunku spadku wartości funkcji. W ten sposób wyznaczamy ciąg kolejnych przybliżeń

$$\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots$$

poszukiwanego minimum.

Procedurę iteracyjną przerywamy, gdy spełniony jest jeden z warunków (warunek **STOPU**):

a)  $\|\mathbf{x}^{i+1} - \mathbf{x}^i\| < \varepsilon$  (najczęściej stosowany)

b)  $\nabla f(\mathbf{x}) = \vec{0}$  (problematyczny w zastosowaniu - jak rozumieć „0”???)

c) lub w kolejnych iteracjach rośnie wartość normy

$$\|\mathbf{x}^{i+1} - \mathbf{x}^i\|$$

co oznacza brak zbieżności - algorytm trzeba zrestartować i/lub zmienić parametry iteracji

## Metoda największego spadku

Korzystamy z pochodnej kierunkowej funkcji w punkcie  $\mathbf{x}'$

$$\left. \frac{df(\mathbf{x}')}{d\lambda} \right|_{\mathbf{u}} = \frac{dF(\lambda)}{d\lambda} = \nabla^T f(\mathbf{x}') \mathbf{u}$$

**Wektor kierunkowy  $\mathbf{u}$  ma długość równą 1**

$$\|\mathbf{u}\| = 1$$

Korzystamy z nierówności Schwartza

$$\nabla^T f(\mathbf{x}') \mathbf{u} \geq -\|\nabla^T f(\mathbf{x}')\| \cdot \|\mathbf{u}\| = -\|\nabla^T f(\mathbf{x}')\| \cdot 1$$

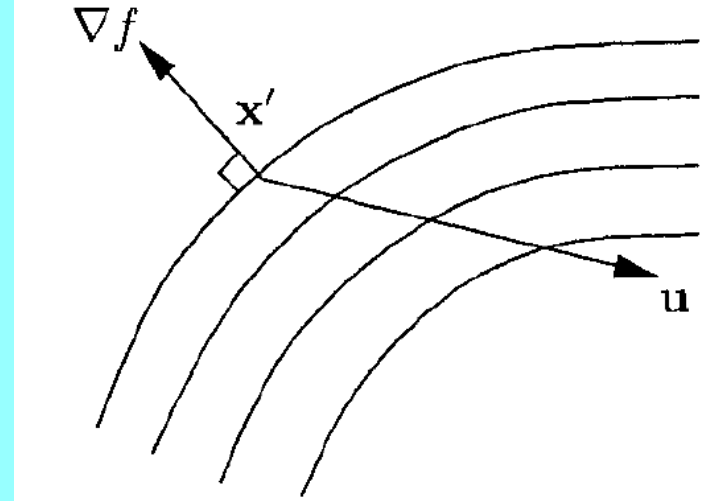
Jeśli wektor kierunkowy wybierzemy w postaci

$$\mathbf{u} = \frac{-\nabla f(\mathbf{x}')}{\|\nabla f(\mathbf{x}')\|}$$

to będzie on wskazywał **kierunek największego spadku**.

Pochodna kierunkowa osiąga wtedy najmniejszą wartość:

$$\frac{dF(\lambda)}{d\lambda} = -\nabla^T f(\mathbf{x}') \frac{\nabla f(\mathbf{x}')}{\|\nabla f(\mathbf{x}')\|} = -1$$



**Rys.** Gradient funkcji i kierunek poszukiwań w metodzie największego spadku.

Algorytm metody największego spadku:

1) Wybierz  $\mathbf{x}^0$

2) Iteracyjnie obliczaj

$$\mathbf{u}^i = \frac{-\nabla f(\mathbf{x}^{i-1})}{\|\nabla f(\mathbf{x}^{i-1})\|}$$

$$\mathbf{x}^i = \mathbf{x}^{i-1} + \lambda \mathbf{u}^i$$

$$F(\lambda_i) = f(\mathbf{x}^{i-1} + \lambda_i \mathbf{u}^i) = \min_{\lambda} f(\mathbf{x}^{i-1} + \lambda \mathbf{u}^i)$$

np.: metoda **złotego podziału** dla

$$\lambda \in (0, \lambda_{max})$$

3) Warunki zakończenia obliczeń

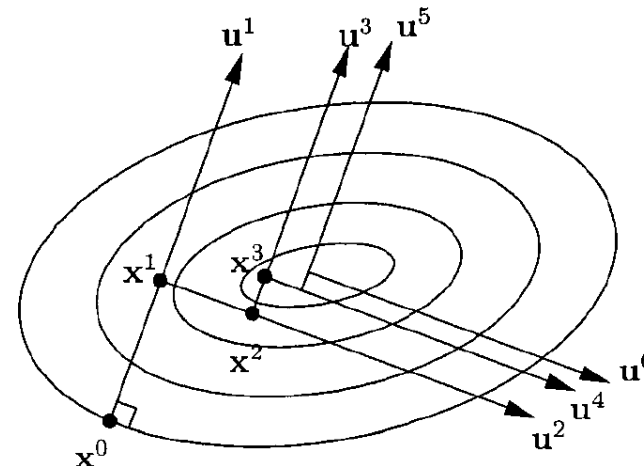
$$\|\mathbf{x}^{i+1} - \mathbf{x}^i\| < \varepsilon_1$$

$$\|\nabla f(\mathbf{x}^i)\| < \varepsilon_2$$

$$|f(\mathbf{x}^i) - f(\mathbf{x}^{i-1})| < \varepsilon_3$$

**Metoda największego spadku może być mało wydajna, jeśli kontur wartości funkcji celu jest wydłużony (elipsa).**

**Pojawiają się wówczas częste zmiany kierunków poszukiwań - zigzag.**

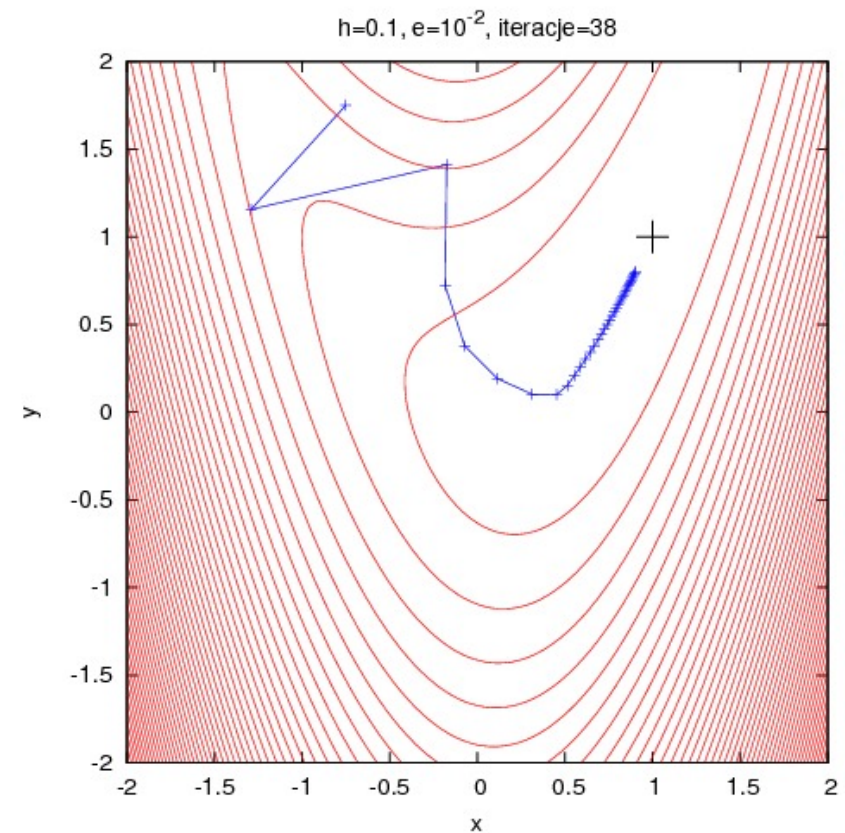
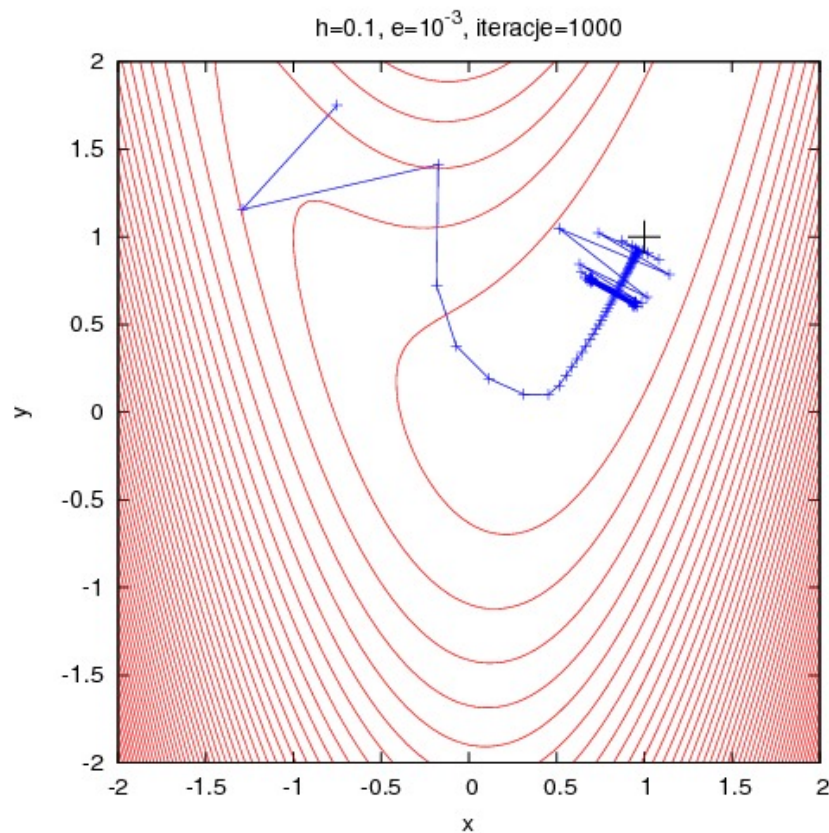


Przykład.

$$f(x, y) = \frac{5}{2}(x^2 - y)^2 + (1 - x)^2$$

Gradient możemy obliczyć  
analitycznie

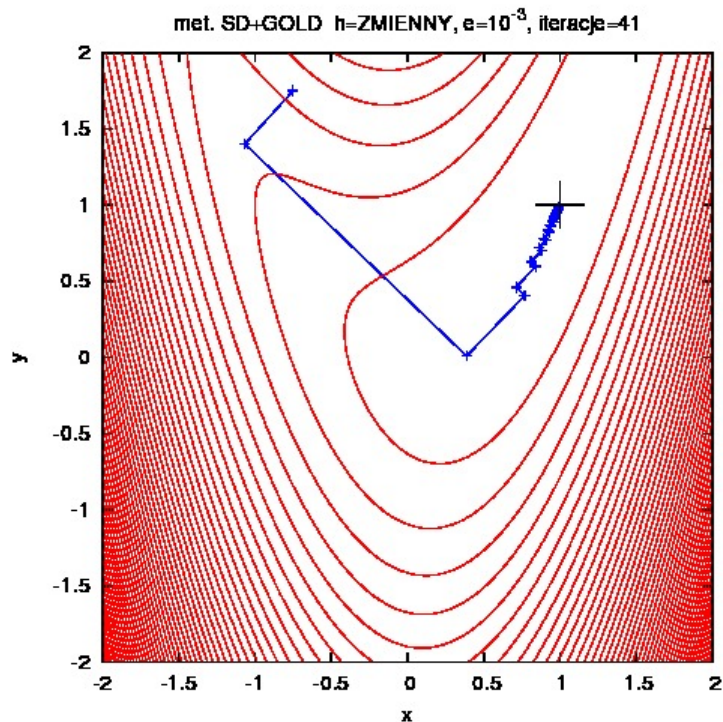
$$\nabla f = \begin{pmatrix} 10(x^2 - y)x + 2x - 2 \\ -5x^2 + 5y \end{pmatrix}$$





Metoda **największego spadku** z optymalizacją parametru  $\lambda$

$$f(x, y) = \frac{5}{2}(x^2 - y)^2 + (1 - x)^2$$



„Tylko” 41 iteracji – ale mamy **zygzak**.

$$\begin{aligned} \frac{\partial f(\mathbf{x}_i + \lambda \mathbf{u}_i)}{\partial \lambda} &= \frac{\partial f(\mathbf{x}_i + \lambda \mathbf{u}_i)}{\partial \mathbf{x}} \mathbf{u}_i \\ &= (\nabla f(\mathbf{x}_i + \lambda \mathbf{u}_i))^T \cdot \mathbf{u}_i = 0 \end{aligned}$$

$$\nabla f(\mathbf{x}_i + \lambda \mathbf{u}_i) = \nabla f(\mathbf{x}_{i+1}) = -\mathbf{u}_{i+1}$$

$$\mathbf{u}_{i+1}^T \cdot \mathbf{u}_i = 0$$

Kierunki poszukiwań w dwóch kolejnych iteracjach są ortogonalne. Jest to duży problem w przypadku, gdy funkcja ma wydłużony kształt w jednym z kierunków (poziomice funkcji są równoległe w szerokim zakresie).

## Metoda sprzężonego gradientu (CG)

Metoda jest wydajna i dostarcza rozwiązania w skończonej liczbie kroków dla problemu w postaci

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T A \mathbf{x} + \mathbf{x}^T \mathbf{b} + c$$

Dwa wektory są wzajemnie sprzężone jeśli

$$\mathbf{u}^T A \mathbf{v} = (\mathbf{u}, A \mathbf{v}) = 0$$

gdzie: A jest macierzą dodatniookreśloną

Jeśli ciąg wektorów

$$\mathbf{u}_i \in R^n, \quad i = 1, 2, \dots, n$$

tworzą wektory wzajemnie sprzężone, to stanowią one bazę w przestrzeni  $R^n$  i wówczas każdy wektor

$$\mathbf{x}_i \in R^n$$

$\alpha_i$  - istnieje, ale nie znamy x-a który pozwoliłby je wyznaczyć

można rozwinąć w tej bazie

$$\mathbf{x} = \sum_{k=1}^n \alpha_k \mathbf{u}_k = \mathbf{x}_0 + \sum_{i=1}^n \lambda_i \mathbf{u}_i$$



$$\alpha_i = \frac{(\mathbf{u}_i, A \mathbf{x})}{(\mathbf{u}_i, A \mathbf{u}_i)}$$

## Algorytm CG Fletchera-Reevsa

1) Wybieramy punkt startowy  $\mathbf{x}^0$  i obliczamy iteracyjnie

$$\mathbf{x}^i = \mathbf{x}^{i-1} + \lambda_i \mathbf{u}_i$$

przy czym kierunek poszukiwań wybieramy jako:

a)  $i=0$

$$\mathbf{u}^1 = -\nabla f(\mathbf{x}^0)$$

b)  $i=1,2,3,..,n$

$$\mathbf{u}^{i+1} = -\nabla f(\mathbf{x}^i) + \beta_i \mathbf{u}^i$$

2) Wartość  $\lambda_i$  wyznaczamy identycznie jak w metodzie największego spadku. **Generowane wektory  $\mathbf{u}^i$  wyznaczają bazę A-ortogonalną.**

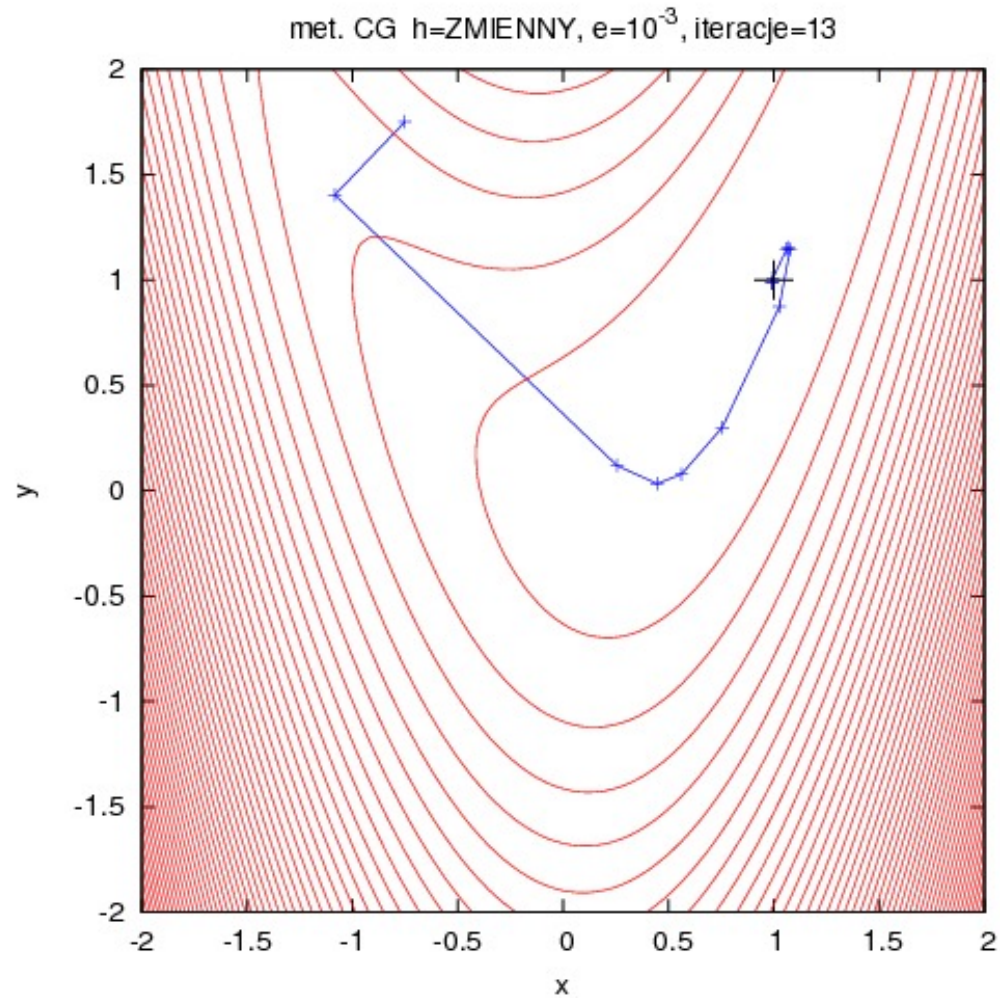
Natomiast parametr  $\beta_i$

$$\beta_i = \frac{\|\nabla f(\mathbf{x}^i)\|^2}{\|\nabla f(\mathbf{x}^{i-1})\|^2}$$

silnie ogranicza zmiany kierunku poszukiwania rozwiązania w kolejnej iteracji (eliminuje „zygzak”).

Przykład. **Metoda CG**

$$f(x, y) = \frac{5}{2}(x^2 - y)^2 + (1 - x)^2$$



Tylko 13 iteracji dzięki brakowi „zygzaku”

## Metoda Newtona poszukiwania minimum funkcji kwadratowej w $R^n$

**Funkcję kwadratową** definiujemy następująco

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T A \mathbf{x} + \mathbf{x}^T \mathbf{b} + c$$

gdzie:  $A$  jest pewną macierzą kwadratową oraz

$$\mathbf{x}, \mathbf{b} \in R^n \quad c \in R$$

Jeśli macierz  $A$  jest symetryczna to wówczas zachodzi

$$\nabla f(\mathbf{x}) = A \mathbf{x} + \mathbf{b}$$

oraz

$$\nabla^2 f(\mathbf{x}) = A$$



$$H(\mathbf{x}) = A$$

(hesjan)

Jeśli  $A$  jest dodatniookreślona to rozwiązanie można łatwo znaleźć, ponieważ

$$\nabla f(\mathbf{x}) = A \mathbf{x} + \mathbf{b} = 0$$

$$\mathbf{x}^* = -A^{-1} \mathbf{b}$$

(macierz dodatniookreślona jest niesobliwa i można ją odwrócić)

W metodzie Newtona zakładamy

$$\mathbf{x}^* = \mathbf{x}^i + \delta$$

gdzie:  $\mathbf{x}^i$  – przybliżone rozwiązanie w i-tej iteracji

Korzystając z rozwinięcia funkcji w szereg Taylora możemy zapisać

$$\begin{aligned} \mathbf{0} &= \nabla f(\mathbf{x}^*) = \nabla f(\mathbf{x}^i + \delta) \\ &= \nabla f(\mathbf{x}^i) + H(\mathbf{x}^i)\delta + O(\|\delta\|^2) \end{aligned}$$

Jeśli pominiemy wyrazy rzędu  $\|\delta\|^2$  to

$$\nabla f(\mathbf{x}^i) + H(\mathbf{x}^i)\delta = 0$$

W i-tej iteracji poprawiamy rozwiązanie, tj.

$$\mathbf{x}^{i+1} = \mathbf{x}^i + \delta$$

i ostatecznie

$$\mathbf{x}^{i+1} = \mathbf{x}^i - H^{-1}(\mathbf{x}^i)\nabla f(\mathbf{x}^i)$$

Oczekujemy, że metoda Newtona będzie pracować również dla innych funkcji niż kwadratowe tj. gdy badaną funkcję celu można lokalnie przybliżyć funkcją kwadratową.

Wadą metody jest konieczność wyznaczania hesjanu w każdym punkcie. Gdy ta staje się osobliwa wówczas metoda przestaje działać – co może być spowodowane np. występowaniem błędów numerycznych.

**Przypadek szczególny** - zastosujemy schemat do funkcji kwadratowej

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}A^T\mathbf{x} + \mathbf{b}^T\mathbf{x} + c$$

W pierwszej iteracji dostajemy

$$\begin{aligned}\mathbf{x}^1 &= \mathbf{x}^0 - A^{-1}(A\mathbf{x}^0 + \mathbf{b}) \\ &= \mathbf{x}^0 - \mathbf{x}^0 - A^{-1}\mathbf{b} = -A^{-1}\mathbf{b}\end{aligned}$$

czyli

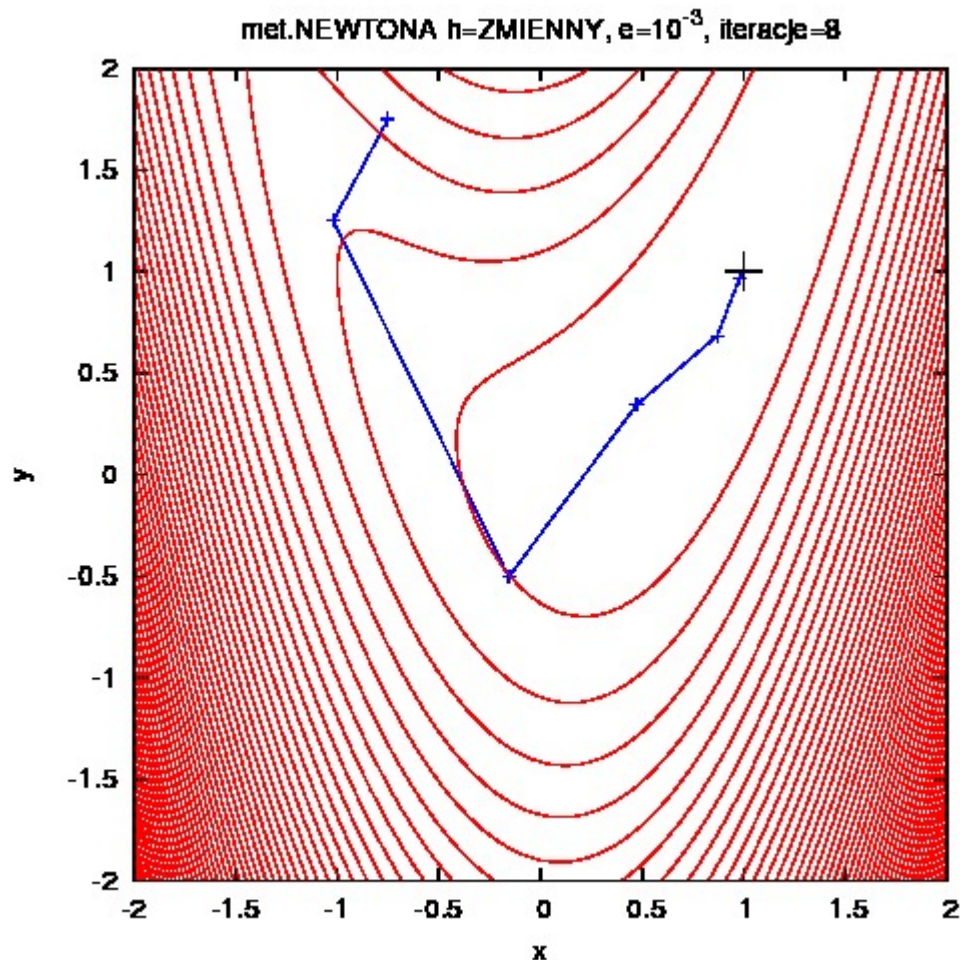
$$\mathbf{x}^1 = \mathbf{x}^*$$

- **zbieżność do minimum uzyskujemy w jednym kroku.**

Ale metoda ma wady:

- 1) Metoda nie zawsze jest zbieżna nawet w pobliżu minimum
- 2) Wymaga znalezienia  $A^{-1}$  w każdej iteracji  
(problem z liczeniem pochodnych, jeśli nie analitycznie to numerycznie  
- wówczas w sposób przybliżony)

## Przykład. **Metoda Newtona**



Rys. **Metoda Newtona** w połączeniu z **metodą złotego podziału** użytą w każdej iteracji.

Tylko 8 iteracji – jak dotąd najmniej.

$$f(x, y) = \frac{5}{2}(x^2 - y)^2 + (1 - x)^2$$

$$H(x, y) = \begin{pmatrix} 30x^2 - 10y + 2 & -10x \\ -10x & 5 \end{pmatrix}$$

Hesjan (pochodne) liczony analitycznie  
- zaleta.

**Czy jest dodatniookreślony?**

To zależy od wektora  $(x, y)$   
bo współczynniki nie są stałe.



## Metoda funkcji kary

Jeśli ograniczamy przestrzeń dopuszczalnych rozwiązań czyli nakładamy warunki np.

$$g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, m$$

lub

$$h_j(\mathbf{x}) = 0, \quad j = 1, 2, \dots, r$$

to mamy do czynienia z problemem z ograniczeniami (**więzami**).

**Ograniczenia te chcemy wbudować w nasz algorytm w taki sposób aby znów problem stał się problemem bez ograniczeń - bez konieczności sprawdzania warunków.**

## Metoda funkcji kary zewnętrznej

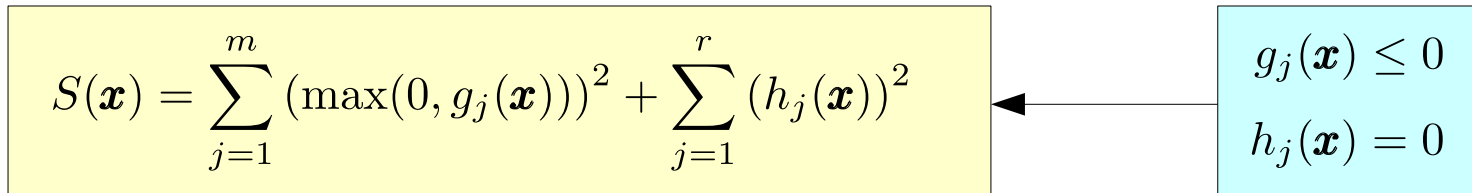
Stosujemy zmodyfikowaną funkcję celu

$$F_i(\mathbf{x}) = f(\mathbf{x}) + c_i S(\mathbf{x})$$

gdzie:  $S(x)$  jest funkcją kary, a  $c_i$  współczynnikiem rosnącym w kolejnych iteracjach

$$\lim_{i \rightarrow \infty} c_i = \infty \quad (c_{i+1} = 2c_i)$$

## Prosty przykład funkcji kary zewnętrznej

$$S(\mathbf{x}) = \sum_{j=1}^m (\max(0, g_j(\mathbf{x})))^2 + \sum_{j=1}^r (h_j(\mathbf{x}))^2$$

$$\begin{aligned} g_j(\mathbf{x}) &\leq 0 \\ h_j(\mathbf{x}) &= 0 \end{aligned}$$

Jeśli spełniony jest pierwszy warunek to pierwsza suma się zeruje. Podobnie jest dla drugiego przypadku. Wtedy

$$S(\mathbf{x}) = 0$$

czyli brak kary w przestrzeni rozwiązań dopuszczalnych. W pozostałym obszarze (**na zewnątrz**), wartość funkcji celu rośnie w kolejnych iteracjach – a więc nie znajdziemy tam minimum.

Zmodyfikowana funkcja celu może nie być różniczkowalna ze względu na funkcję  $\max()$  w definicji funkcji kary co eliminuje metody gradientowe przy poszukiwaniu minimum.

### Sposób postępowania przy użyciu **kary zewnętrznej**:

- 1) wybieramy punkt startowy  $\mathbf{x}_0$
- 2) określamy  $c_i$  oraz modyfikujemy funkcję celu
- 3) znajdujemy rozwiązanie  $\mathbf{x}^*$  przy użyciu wybranej metody
- 4) sprawdzamy przyjęty warunek stopu jak w metodzie bez ograniczeń
- 5) jeśli warunek stopu nie jest spełniony to wykonujemy kolejną iterację przyjmując

$$\mathbf{x}_i = \mathbf{x}^*, c_{i+1} = a \cdot c_i$$

## Metoda funkcji kary wewnętrznej

Funkcja kary wewnętrznej przyjmuje wartości dodatnie wewnątrz obszaru rozwiązań dopuszczalnych, a jej wartość gwałtownie rośnie przy zbliżaniu się do brzegu (bariera).

Zakładamy że warunki ograniczające mają postać

$$g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, m$$

W każdej iteracji modyfikujemy funkcję celu wprowadzając funkcję kary  $S(\mathbf{x})$

$$F_i(\mathbf{x}) = f(\mathbf{x}) + c_i S(\mathbf{x})$$

z warunkiem dla współczynnika  $c_i$

$$\lim_{i \rightarrow \infty} c_i = 0 \quad (c_{i+1} = a \cdot c_i, \quad a \in (0, 1))$$

Własności funkcji kary wewnętrznej:

a) 
$$g_j(\mathbf{x}) < 0, \quad j = 1, \dots, m \Rightarrow S(\mathbf{x}) > 0$$

b) jeśli dla ciągu przybliżeń  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_i$  spełniony jest warunek

$$g_j(\mathbf{x}_i) < 0, \quad j = 1, \dots, m, \quad i = 1, 2, \dots$$

oraz dla określonego  $j$

$$\lim_{i \rightarrow \infty} g_j(\mathbf{x}_i) = 0$$

to

$$\lim_{i \rightarrow \infty} S(\mathbf{x}_i) \rightarrow \infty$$

Przykłady funkcji barierowych w metodzie kary wewnętrznej:

$$S(\mathbf{x}) = - \sum_{j=1}^m \frac{1}{g_j(\mathbf{x})}$$

$$S(\mathbf{x}) = - \sum_{j=1}^m \ln(-g_j(\mathbf{x}))$$

### Sposób postępowania:

- 1) wybieramy punkt startowy  $\mathbf{x}_0$
- 2) określamy  $c_i$  oraz modyfikujemy funkcję celu
- 3) znajdujemy rozwiązanie  $\mathbf{x}^*$  przy użyciu wybranej metody
- 4) sprawdzamy przyjęty warunek stopu oraz dodatkowo sprawdzamy zgodność z warunkami ograniczającymi

$$c_i |S(\mathbf{x}_i)| < \varepsilon$$

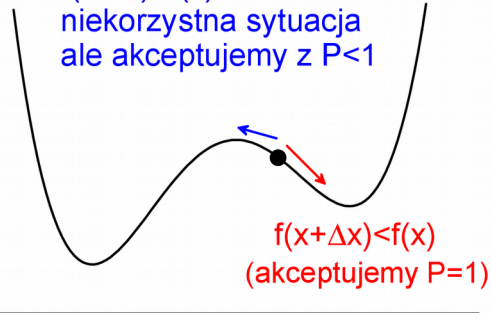
- 5) jeśli warunki stopu nie są spełnione to wykonujemy kolejną iterację przyjmując

$$\mathbf{x}_i = \mathbf{x}^*, c_{i+1} = a \cdot c_i$$

# Metoda symulowanego wyżarzania (jedna z metod Monte Carlo, metoda stochastyczna)

1) start

$f(x-\Delta x) > f(x)$   
niekorzystna sytuacja  
ale akceptujemy z  $P < 1$



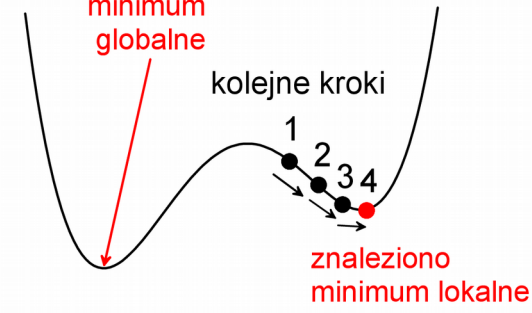
2) akceptujemy tylko kroki z  $P=1$   
co si4 stanie?

szukane  
minimum  
globalne

kolejne kroki

1  
2  
3  
4

znaleziono  
minimum  
lokalne



Prawdopodobieństwo akceptacji  
„gorszego” rozwiązania

$$P = \exp\left(-\frac{f(x_i + \Delta x) - f(x_i)}{T}\right)$$

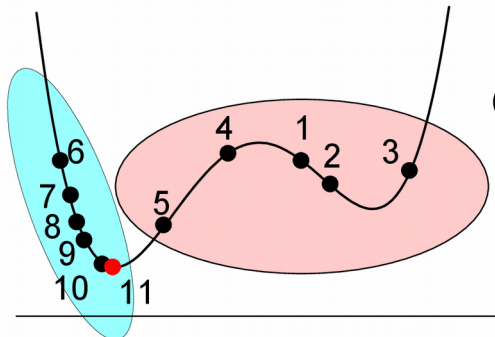
zależy silnie od parametru  $T$   
(w fizyce jest to temperatura).

$T$  duże: wartość wykładnika bliska  
zeru ( $P \rightarrow 1$ )

$T$  małe: wykładnik jest dużą ujemną  
liczbą ( $P \rightarrow 0$ )

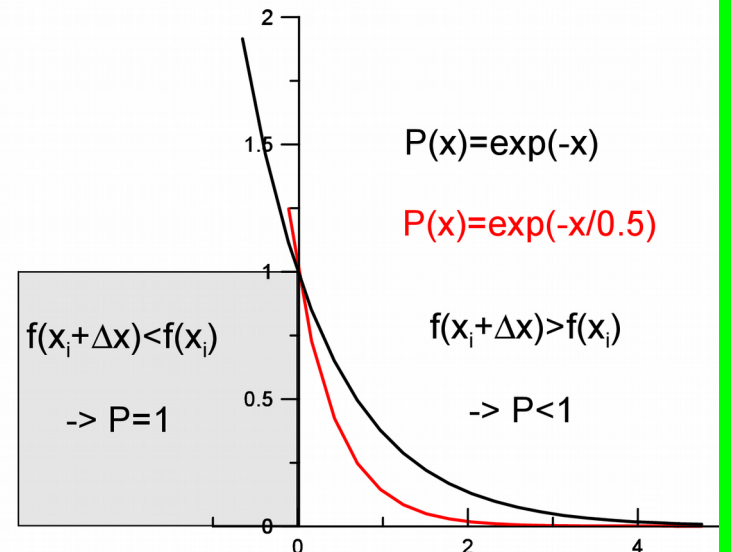
3) akceptujemy kroki z  $P=1$  oraz  $P < 1$   
i zmieniamy (obniżamy) temperaturę

w kolejnych iteracjach  
obniżamy temperaturę  
(małe przemieszczenia)



początek symulacji  
wysoka temperatura  
(duże przemieszczenia)

mając szczęście za  $N$ -tym  
wywołaniem algorytmu  
znajdziemy minimum globalne  
(ale nie zawsze tak jest)



## Najprostszy algorytm symulowanego wyżarzania - iteracyjny

Start: wybieramy punkt startowy  $x_0$  i określamy wartość funkcji celu  $f(x_0)$  dla naszego wędrowca oraz ustalamy temperaturę  $T$  i ilość kroków  $M$  jaką ma wykonać

1) losujemy przemieszczenie  $\Delta x$  generatorem liczb pseudolosowych i określamy  $f(x_i + \Delta x)$

2) akceptujemy nowe położenie z prawdopodobieństwem  $P=1$  jeśli

$$f(x_i + \Delta x) \leq f(x_i) \Rightarrow x_{i+1} = x_i + \Delta x$$

w przeciwnym wypadku

$$f(x_i + \Delta x) > f(x_i)$$

określamy prawdopodobieństwo

$$P = \exp\left(-\frac{f(x_i + \Delta x) - f(x_i)}{T}\right) < 1$$

i losujemy liczbę

$$X \in U(0, 1)$$

nowe położenie akceptujemy (z prawdopodobieństwem  $P$ ) jeśli spełniona jest relacja

$$X < P$$

3) co  $M$  iteracji obniżamy temperaturę np.

$$T_{new} = C \cdot T_{old}, \quad C < 1$$

co efektywnie zmniejsza prawdopodobieństwo akceptacji położenia o wyższej funkcji kosztu

**Kroki 1-3 powtarzamy  $M$  razy dla każdego wędrowca.  
Symulację możemy wykonać dla  $N$  wędrowców równoległe - są niezależni.  
Po jej zakończeniu wyszukujemy wędrowca o najniższej funkcji celu.**

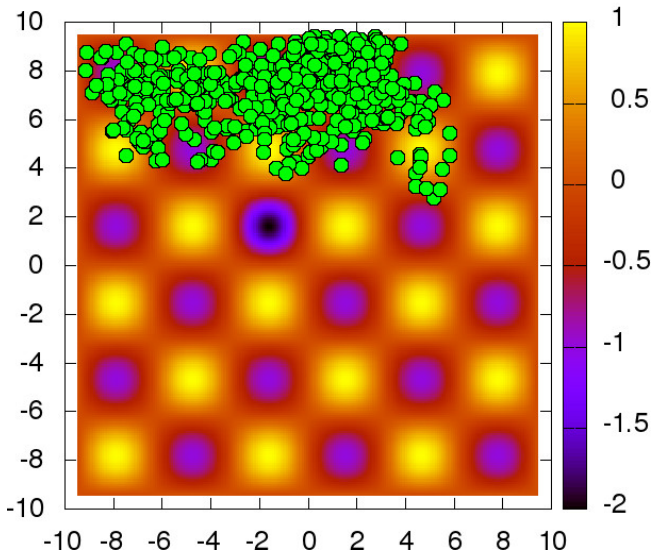
**Przykład.** Szukamy minimum globalnego funkcji celu

$$f(x, y) = \sin(x) \sin(y) - \exp\left(-\left(x + \pi/2\right)^2 - \left(y - \pi/2\right)^2\right)$$

$$x, y \in [-3\pi, 3\pi]$$

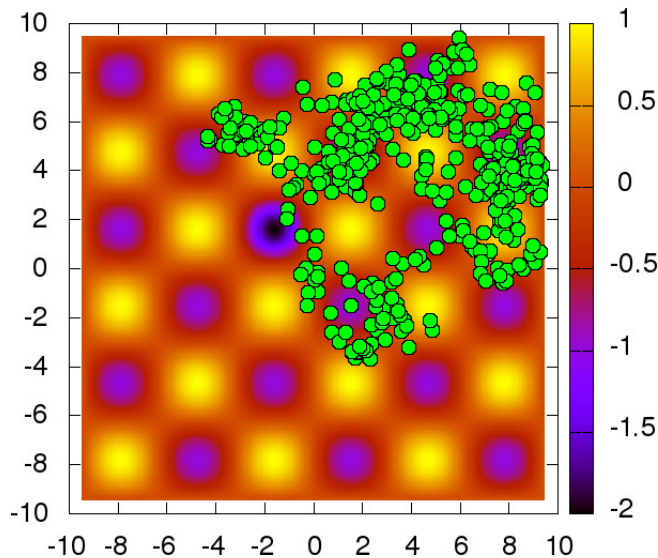
Wynik dla pojedynczego wywołania algorytmu (jeden wędrowiec) - kropki to kolejne położenia.  
Dla każdego przypadku - start z tego samego punktu.

start=(5,5), T=10, 500 krokow



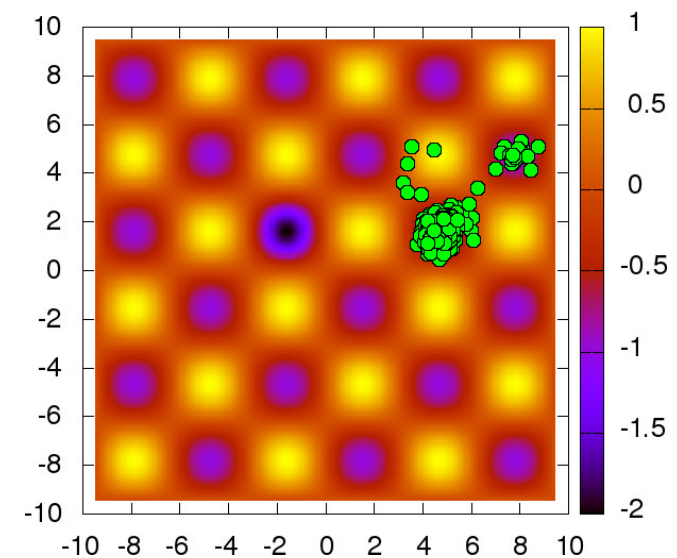
Temperatura za duża  
- zbyt duży krok,  
(wędrowiec odwiedza  
maksima)

start=(5,5), T=1, 500 krokow



Temperatura średnia  
(wędrowiec omija maksima  
przemieszcza się dolinami)

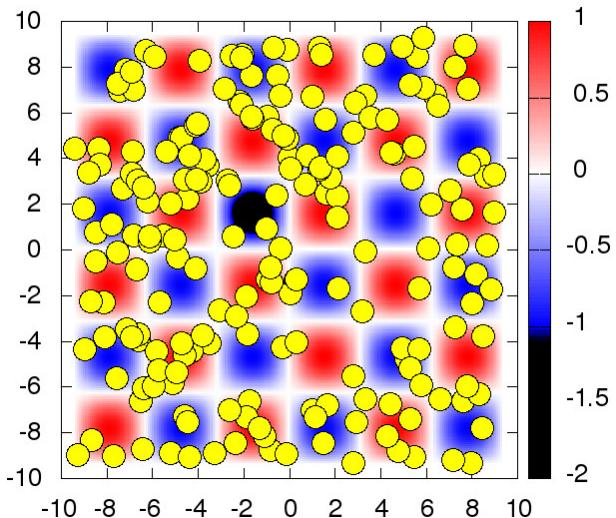
start=(5,5), T=0.1, 500 krokow



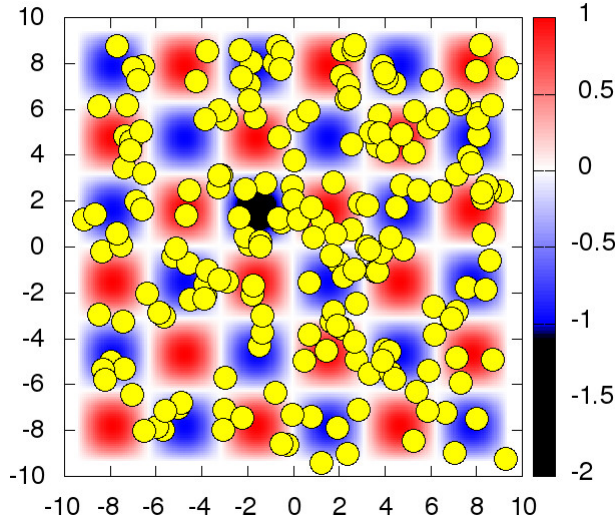
Temperatura niska  
(wędrowiec bardzo szybko  
lokalizuje się w minimum)

Wyniki dla  $M=200$  wędrówców i temperatury obniżanej co 100 iteracji ( $T_{\text{new}} = T_{\text{old}}/2$ )  
Metodę łatwo zrównoleglić.

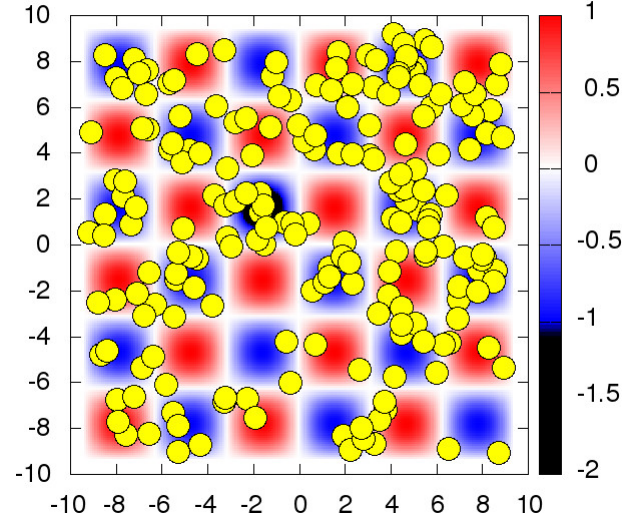
$N=200, dx, dy < 1, T=T_0/2^0$



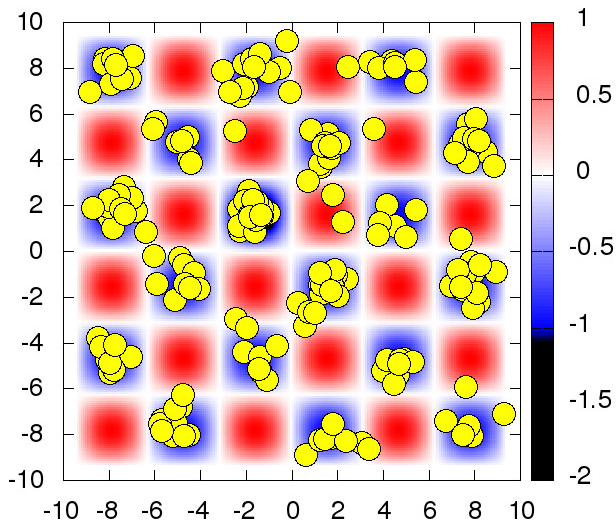
$N=200, dx, dy < 1, T=T_0/2^2$



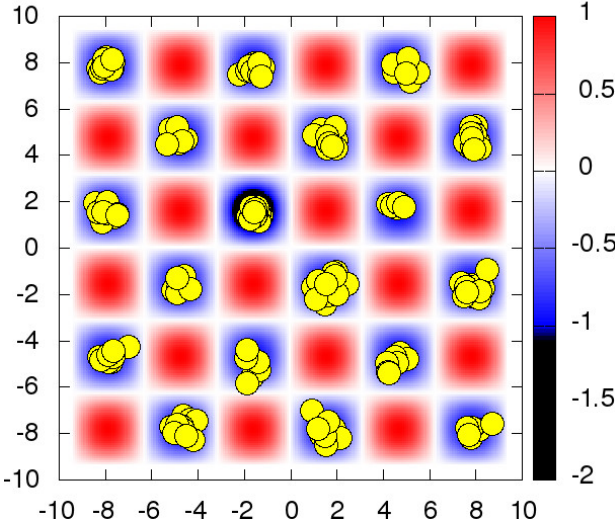
$N=200, dx, dy < 1, T=T_0/2^4$



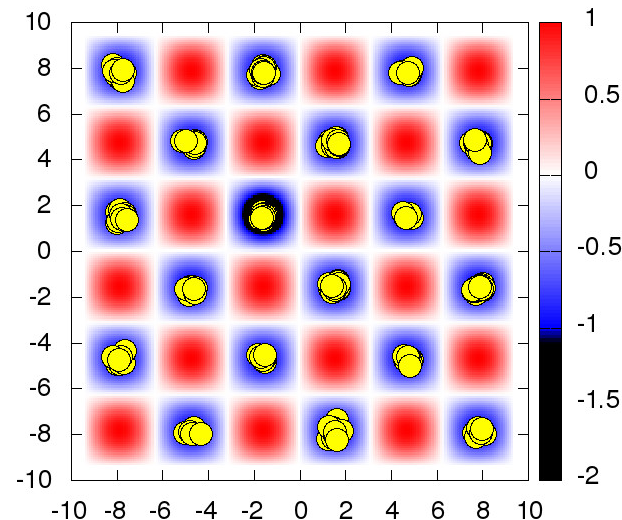
$N=200, dx, dy < 1, T=T_0/2^6$



$N=200, dx, dy < 1, T=T_0/2^8$



$N=200, dx, dy < 1, T=T_0/2^{10}$



Jedno ze znalezionych minimów  
to minimum globalne.