Ukázky optimalizačních algoritmů na funkci dvou proměnných se třemi minimy

Tomáš Kroupa

20. května 2014

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Abstrakt

V následujícím textu bude na jednoduché úloze minimalizace funkce s jedním globálním minimem a dvěma lokálními ukázáno jak fungují jednotlivé typy algoritmů pro hledání zmíněného minima.

Tento studijní materiál je spolufinancován Evropským sociálním fondem a státním rozpočtem České republiky.

Obsah

Tento studijní materiál je spolufinancován Evropským sociálním fondem a státním rozpočtem České republiky.

1 Formulace úlohy

Je dána funkce dvou proměnných

$$
\begin{aligned}
r = r(x,y) = & -30e^(-0.03(1((x+1)^2)+3((y-4)^2))) \\
& -20e^(-0.02(2((x+8)^2)+1((y+5)^2))) \\
& -10e^(-0.01(3((x-6)^2)+2((y+7)^2))) \\
& +100
\end{aligned} \tag{1}
$$

Tato funkce je definovaná na čtvercové oblasti, kde

$$x \in \langle -20, 20 \rangle \text{ a} \tag{2}$$
$$y \in \langle -20, 20 \rangle . \tag{3}$$

Úkolem v dalším textu ukázaných optimalizačních algoritmů je nalézt minimum funkce r, které je globálním minimem a to v bodě

$$x = -1 \text{ a} \tag{4}$$
$$y = 4. \tag{5}$$

V definované oblasti existuje právě jedno globální minimum a dvě lokální

$$x = -8 \text{ a} \tag{6}$$
$$y = -5 \tag{7}$$

a

$$x = 6 \text{ a} \tag{8}$$
$$y = -7. \tag{9}$$

Poznámky V praxi a také v užívaném softwaru OptiSLang se funkce r nazývá objektivní funkcí, nebo minimalizovanou funkcí. V analýze, která je v tomto ukázkovém případě pouhým výpočtem hodnoty funkce r, může v obecném (reálnějším) případě být vypočteno více různých hodnot tzv. výstupních parametrů. Z nich lze následně vyčíslit objektivní funkci. Hodnoty x a y se nazývají vstupní parametry, těch může být opět větší množství. V reálných problémech nemusí navíc jít o spojitě se měnící parametry, ale i o diskrétní hodnoty.

Jednotlivé výpočty objektivní funkce budeme nazývat designy. Protože výpočet, zejména při využití metody konečných prvků, jednotlivých designů může trvat i několik hodin, je důležité vždy vybrat vhodný algoritmus, který dosáhne minima při co nejmenším počtu designů.

2 Informace k následujícím kapitolám

V následujících kapitolách je ukázáno fungování množství algoritmů. K textu je vhodné použít i animace, které ukazují, jak jednotlivé algoritmy postupují během optimalizace.

3 Popis co se děje v animacích a co a jak je nastaveno

3.1 Gradientní

3.1.1 LBFGS – Počáteční bod $[x_0, y_0] = [-20, -20]$

Nastavení výpočtu

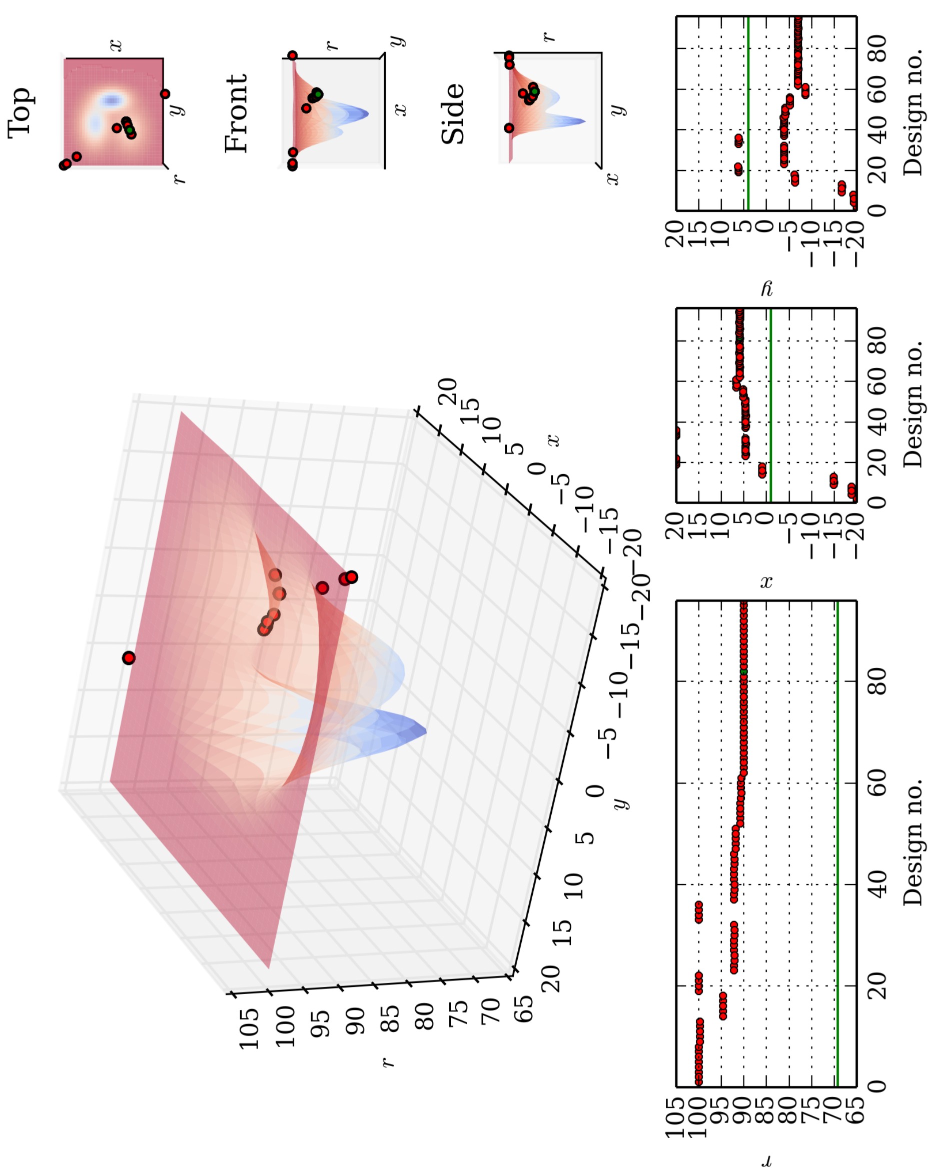
- Maximální počet iterací (maximum naubmer of iterations) = 20
- Maximální počet volání funkce (maximum number of function calls) = 20
- Velikost intervalu pro výpočet gradientu (size of differentiation interval) = 0.1%
- Výpočet gradientu (differentiation method) = central differences, Pro výpočet gradientu je potřeba 5 výpočtů residua (designů). Vědy hodnotu uprostřed gradientu a po obou stranách.

Průběh výpočtů

- Hodnota nalezeného minima $r = 89.99$
- Minimum nalezeno v bodě $[x, y] = [5.98, -6.99]$

Komentáře

- Dosaženo lokálního minima



Obrázek 1: Animace: function_3_LBFGS_m20m20_OPTGRAD.avi

3.1.2 LBFGS – Počáteční bod $[x_0, y_0] = [15, 15]$

Nastavení výpočtu

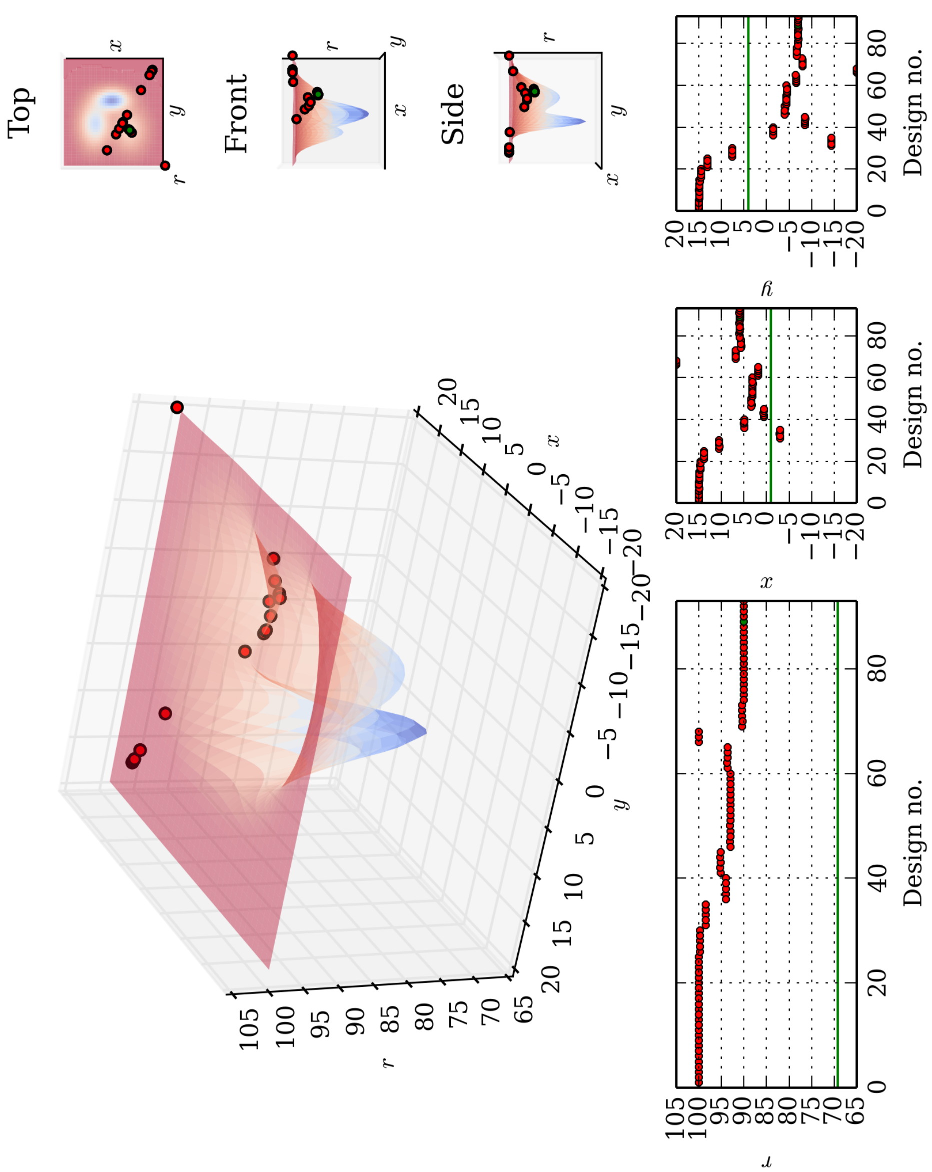
- Maximální počet iterací (maximum naubmer of iterations) = 20
- Maximální počet volání funkce (maximum number of function calls) = 20
- Velikost intervalu pro výpočet gradientu (size of differentiation interval) = 0.1%
- Výpočet gradientu (differentiation method) = central differences, Pro výpočet gradientu je potřeba 5 výpočtů residua (designů). Vědy hodnotu uprostřed gradientu a po obou stranách.

Průběh výpočtů

- Hodnota nalezeného minima $r = 89.99$
- Minimum nalezeno v bodě $[x, y] = [5.98, -6.99]$

Komentáře

- Dosaženo lokálního minima
- Jiný počáteční bod, ale stejný výsledek.



Obrázek 2: Animace: function_3_LBFGS_1515_OPTGRAD.avi

3.1.3 NLPQLP – Počáteční bod $[x_0, y_0] = [-20, -20]$

Nastavení výpočtu

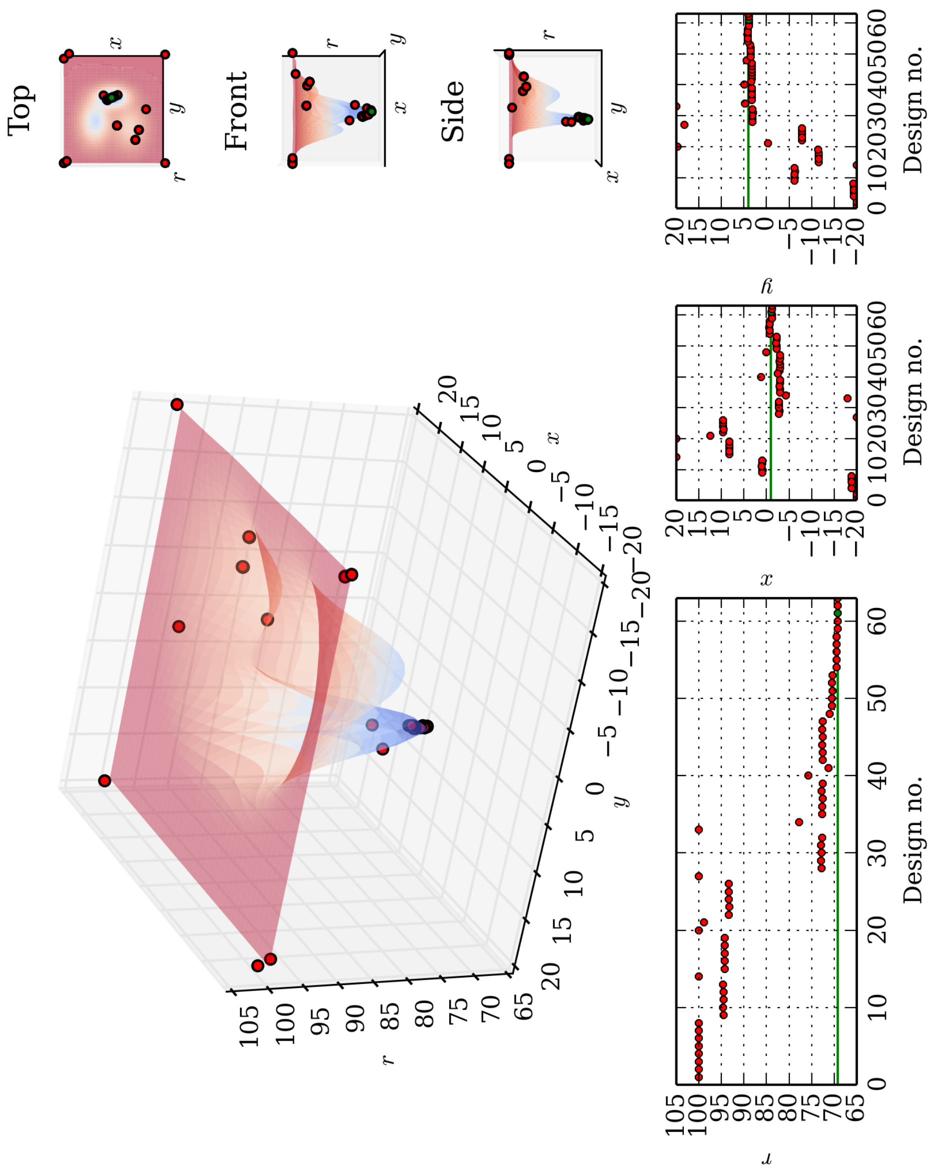
- Maximální počet iterací (maximum naubmer of iterations) = 20
- Maximální počet volání funkce (maximum number of function calls) = 20
- Velikost intervalu pro výpočet gradientu (size of differentiation interval) = 0.1%
- Výpočet gradientu (differentiation method) = central differences, Pro výpočet gradientu je potřeba 5 výpočtů residua (designů). Vědy hodnotu uprostřed gradientu a po obou stranách.

Průběh výpočtů

- Hodnota nalezeného minima $r = 69.21$
- Minimum nalezeno v bodě $[x, y] = [-1.14, 3.93]$

Komentáře

- Dosaženo okolí globálního minima
- Stejný počáteční bod jako v případě LBFGS a jiný výsledek!



Obrázek 3: Animace: function_3_NLPQLP_m20m20_OPTGRAD.avi

3.1.4 NLPQLP – Počáteční bod $[x_0, y_0] = [15, 15]$

Nastavení výpočtu

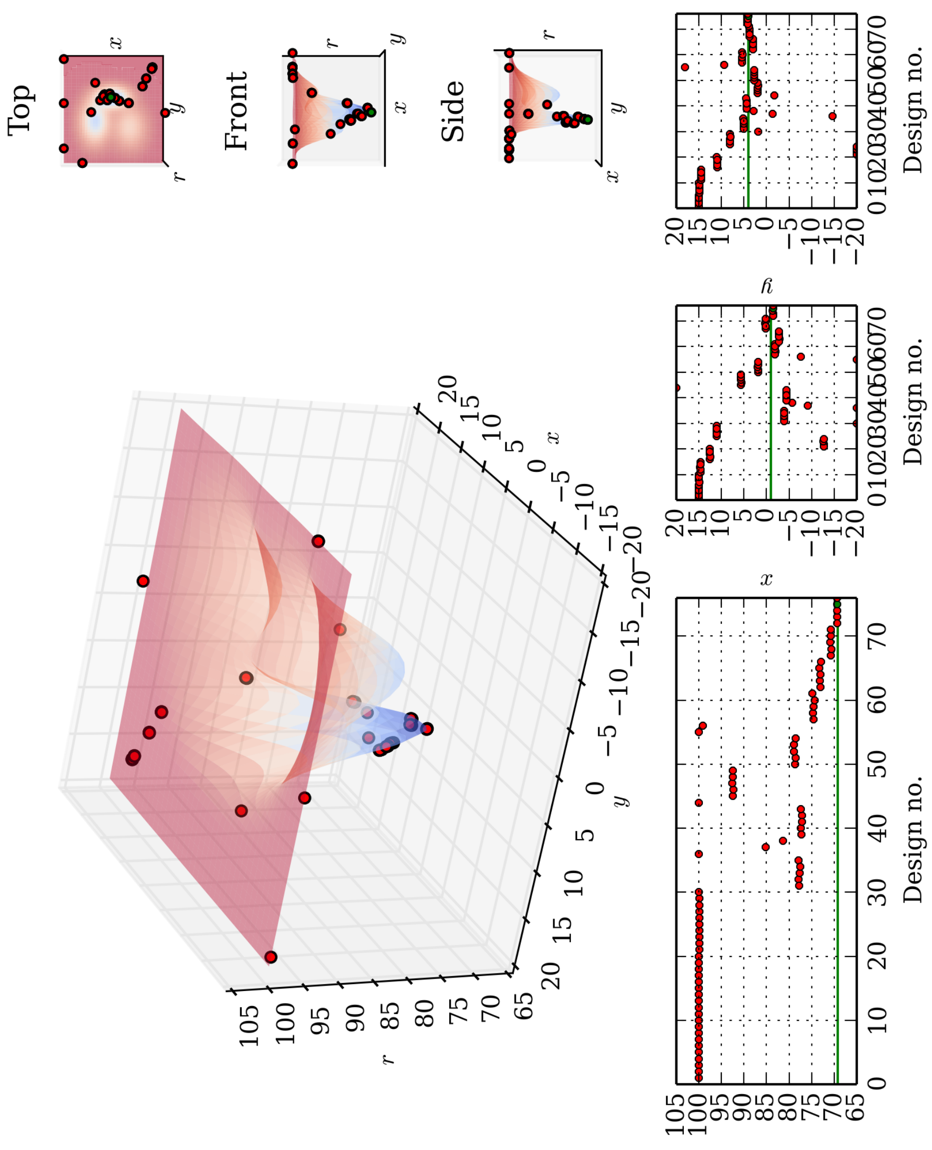
- Maximální počet iterací (maximum naubmer of iterations) = 20
- Maximální počet volání funkce (maximum number of function calls) = 20
- Velikost intervalu pro výpočet gradientu (size of differentiation interval) = 0.1%
- Výpočet gradientu (differentiation method) = central differences, Pro výpočet gradientu je potřeba 5 výpočtů residua (designů). Vědy hodnotu uprostřed gradientu a po obou stranách.

Průběh výpočtů

- Hodnota nalezeného minima $r = 69.32$
- Minimum nalezeno v bodě $[x, y] = [4.06, -1.43]$

Komentáře

- Dosaženo lokálního minima
- Stejný počáteční bod jako v případě LBFGS a jiný výsledek!



Obrázek 4: Animace: function_3_NLPQLP_1515_OPTGRAD.avi

3.2 Optimalizační algoritmy inspirované přírodou

3.2.1 Evolution algorithm

Nastavení výpočtu

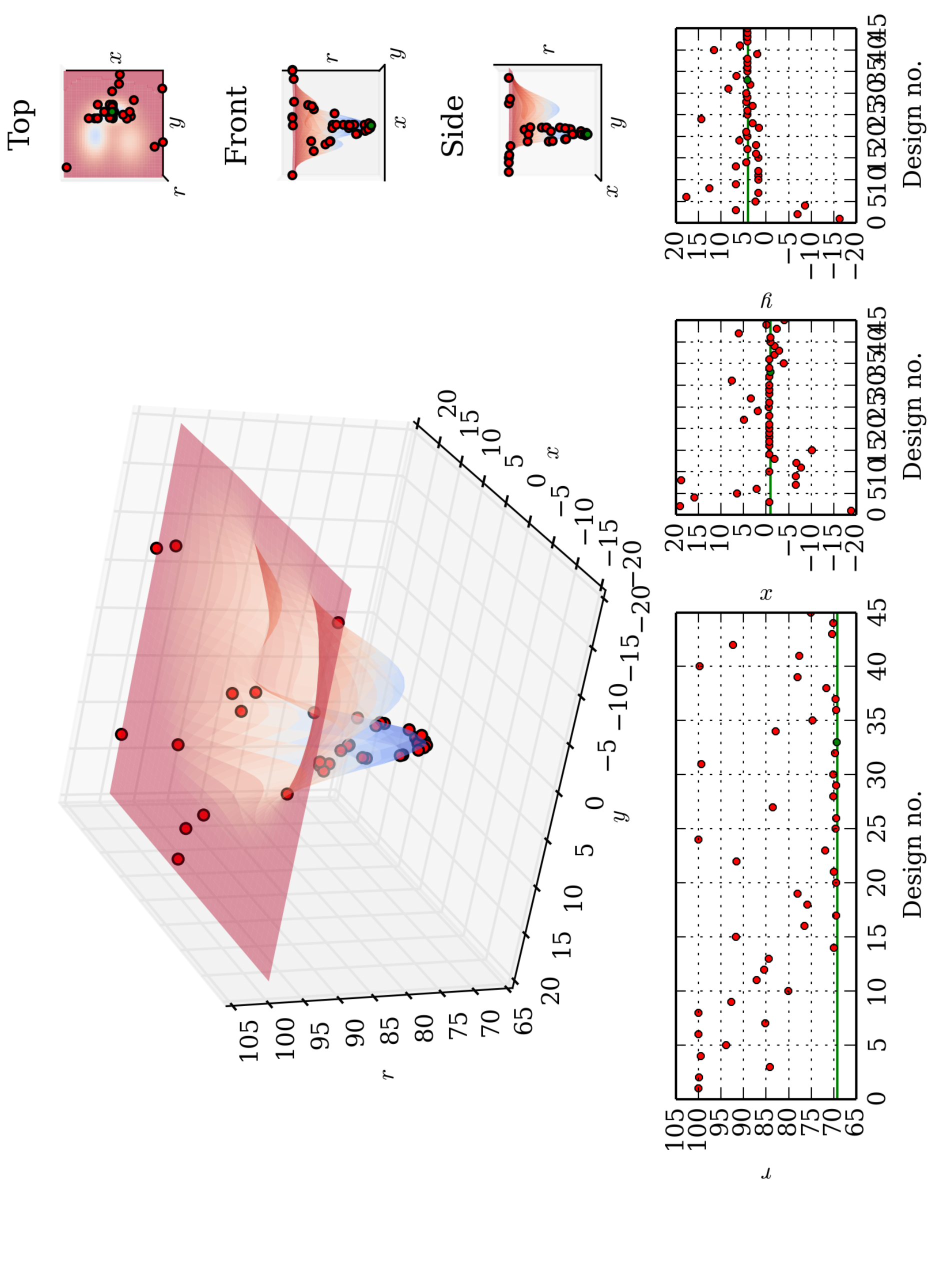
- Počáteční generace (initial generation) = 8
- Minimální počet generací (Minimum number of generations) = 10
- Maximální počet generací (Maximum number of generations) = 23
- Po kolik ageneracích kdy algoritmus nenajde lepší řešení zastavit (Stop after generation of stagnation) = 10
- Velikost archivu (Archiv size) = 4
- Jak známkovat nejlepší řešení (Ranking method) = linear
- Počet rodičů = 4
- Rozptyl mutace na začátku optimalizace (Std. deviation start) = 0.1
- Rozptyl mutace na konci optimalizace (Std. deviation end) = 0.01
- Všechny ostatní nastavení ponechány jako defaultní-doporučené hodnoty

Průběh výpočtů

- Hodnota nalezeného minima $r = 69.3092$
- Minimum nalezeno v bodě $[x, y] = [-0,912, 4.087]$

Komentáře

- Ne všechny designy se počítají, protože výpočet byl nastaven tak, že pokud ve dvou generacích je nutné počítat stejný design (design se stejnými vstupními parametry), tak tento design není počítán z důvodu úspory výpočetního času.



Obrázek 5: Animace: function_3_EA_NOA.avi

3.2.2 Particle swarm

Nastavení výpočtu

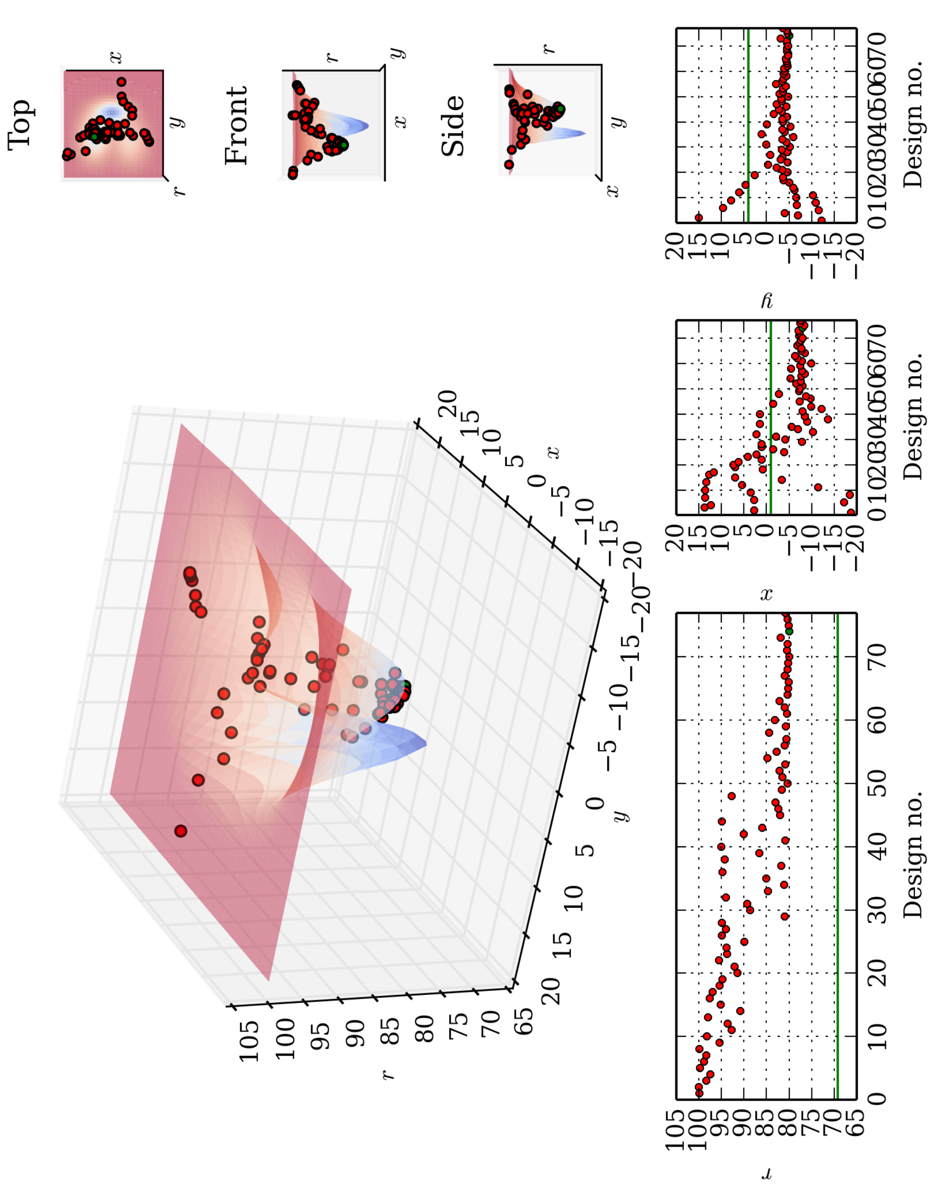
- Počet jedinců v generaci (Population size) = 4
- Minimální počet generací (Minimum number of generations) = 20
- Maximální počet generací (Maximum number of generations) = 20
- Po kolik ageneracích kdy algoritmus nenajde lepší řešení zastavit (Stop after generation of stagnation) = 20
- Prohledavaci strategie (Search strategy) = Global search
- – (Inertia weight) = 0.9 (begin), 0.2 (end)
- – (Personal acceleration) = 0.9 (begin), 0.1 (end)
- – (Global acceleration) = 0.1 (begin), 0.9 (end)
- Typ mutace (Mutation type) = Normal distribution
- Frevence křížení (Mutation rate) = 0.2
- Rozptyl mutace na začátku optimalizace (Std. deviation start) = 0.1
- Rozptyl mutace na konci optimalizace (Std. deviation end) = 0.01

Průběh výpočtů

- Hodnota nalezeného minima $r = 79.973$
- Minimum nalezeno v bodě $[x, y] = [-7.962, -5.088]$

Komentáře

- Dosaženo lokálního minima - krásná ukázka nebezpečnosti využívání „nature based" algoritmů. Samozřejmě tomuto lze zabránít zejména tím, že se nastaví veliký počet jedinců v počáteční generaci, tak aby se prohledal důkladně celý prostor. pak lze počítat s tím, že nejlepší designy leží již v okolí globálního minima.



Obrázek 6: Aniamce: function_3_PSO_NOA.avi

3.2.3 Simple design improvement – start design $[x_0, y_0] = [0, 0]$

Nastavení výpočtu

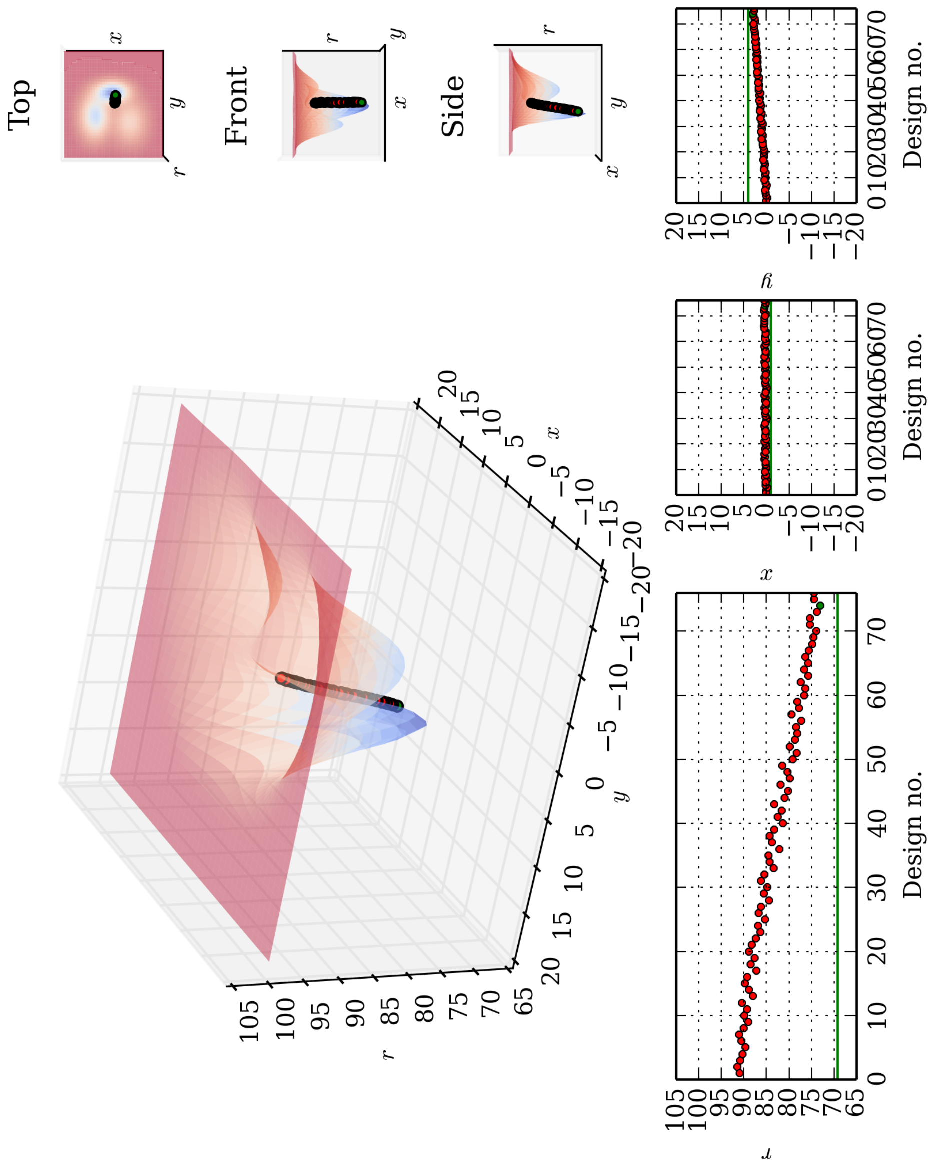
- Počet jedinců v generaci (Number of samples) = 4
- Šířka rozhledu pro hledání dalších lepších výsledků (Sampling bound width) = 1% z (horní mez - dolní mez parametru)
- Minimální počet generací (Minimum number of generations) = 20
- Maximální počet generací (Maximum number of generations) = 20
- Zastavovací podmínka = Type 2 = 0 (Detorioration of performance) – Nula ponechána aby algoritmus běžel pořád

Průběh výpočtů

- Hodnota nalezeného minima $r = 73.112$
- Minimum nalezeno v bodě $[x, y] = [2.983, 0.283]$

Komentáře

- Optimalizace směřuje ke glonbálnímu minimu díky šťastně zvolenému počátečnímu bodu.
- Díky nastavení malého kroku lze sledovat pomalou konvergenci.
- Ani po 80 designech výpočet dedoběhl do minima.



Obrázek 7: Animace: function_3_SDI_00_NOA.avi

3.2.4 Simple design improvement – start design $[x_0, y_0] = [-20, -20]$

Nastavení výpočtu

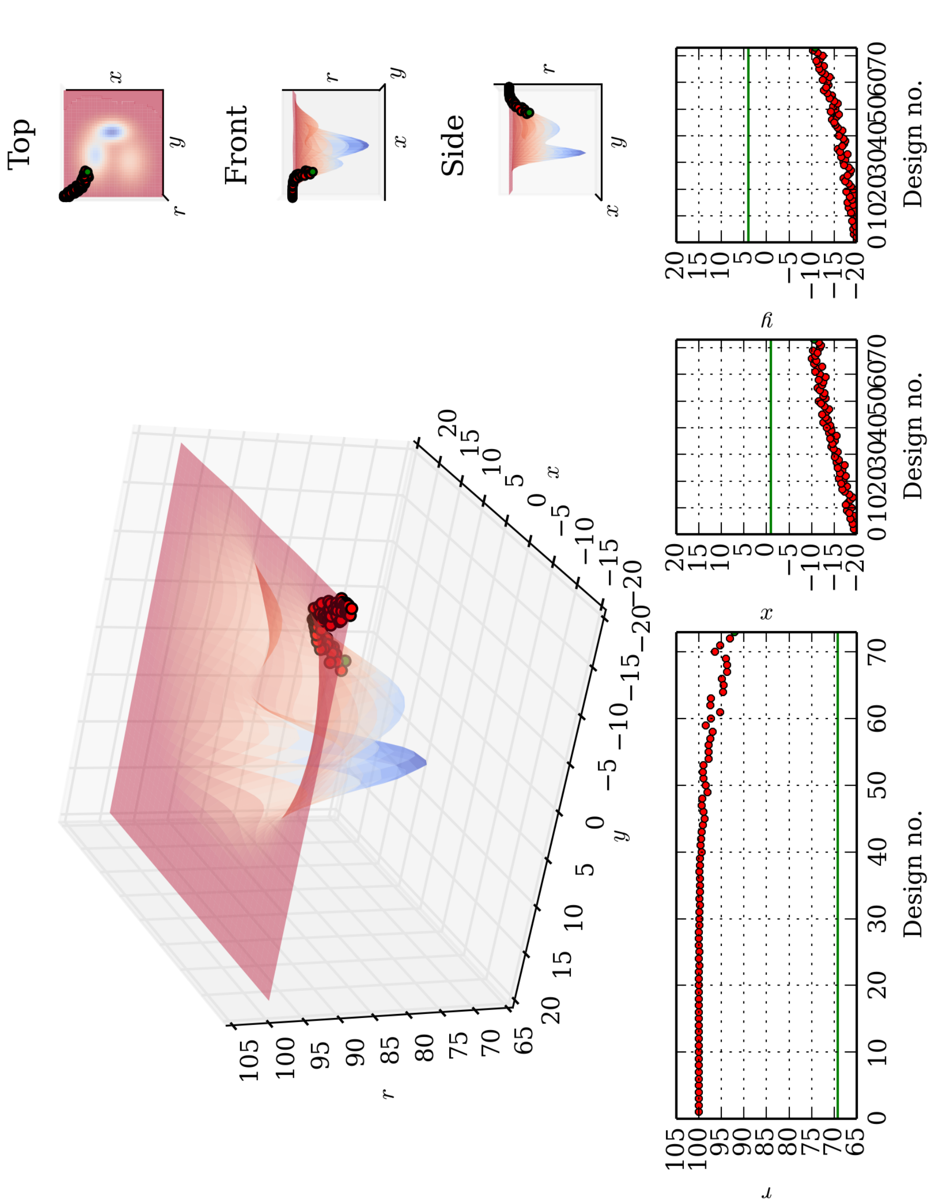
- Počet jedinců v generaci (Number of samples) = 4
- Šířka rozhledu pro hledání dalších lepších výsledků (Sampling bound width) = 5% z (horní mez - dolní mez parametru)
- Minimální počet generací (Minimum number of generations) = 20
- Maximální počet generací (Maximum number of generations) = 20
- Zastavovací podmínka = Type 2 = 0 (Detorioration of performance) – Nula ponechána aby algoritmus běžel pořád

Průběh výpočtů

- Hodnota nalezeného minima $r = 92.092$
- Minimum nalezeno v bodě $[x, y] = [-10.58, -10.75]$

Komentáře

- Optimalizace směřuje k lokálnímu minimu díky nešŤastně zvolůeném u počátečnímu bodu.
- Ani po 80 designech výpočet dedoběhl do minima.



Obrázek 8: Animace: function_3_SDI_m20m20_NOA.avi

Tato prezentace je spolufinancována Evropským sociálním fondem a státním rozpočtem České republiky v rámci projektu č. **CZ.1.07/2.2.00/28.0206**
„Inovace výuky podpořená praxí“.

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Tento studijní materiál je spolufinancován Evropským sociálním fondem a státním rozpočtem České republiky.