

Metody Losowe Optymalizacji Globalnej

Michał Okulewicz

Wydział Matematyki i Nauk Informatycznych
Politechnika Warszawska

Informacje organizacyjne

- 1 Prowadzący
- 2 Zakres przedmiotu
- 3 Zasady zaliczenia
- 4 Literatura
- 5 Zadanie optymalizacji

Prowadzący I

- Studia: magisterskie na MiNI (MSI), doktoranckie na IPI PAN (SMAD dla leniwych z modyfikacjami)
- Praca magisterska: *Porównanie wybranych metod optymalizacji globalnej*, 2011
- Rozprawa doktorska: *Zastosowanie populacyjnych metaheurystyk wykorzystujących rozkład danych problemu do rozwiązywania problemu dynamicznej marszrutyzacji*, 2017
- Rozwiązywane obecnie zagadnienia:
 - wizualna atrakcyjność rozwiązań
 - algorytm optymalizacyjnych ogólnego przeznaczenia
- Ulubione algorytmy metaheurystyczne
 - Particle Swarm Optimization,
 - Differential Evolution,

Prowadzący II

- Variable Neighborhood Search
- Inne wykorzystywane techniki:
 - modele liniowe,
 - filtr cząsteczkowy,
 - lasy losowe, regresja logistyczna sieci neuronowe
- Przekonania naukowe:
 - optymalizację rojem cząstek należałoby uogólnić,
 - optymalizacja rojem cząstek oferuje dobrą platformę do rozwoju różnych zachowań cząstek,
 - interakcje ludzi w zespole mogą stanowić dobre źródło inspiracji dla tych zachowań,
 - algorytmy powinny być proste w implementacji, a wydajne tylko na tyle ile potrzeba, bo mojego czasu mi szkoda, a czasu procesora nie,

Prowadzący III

- polski rynek jest gotowy na wdrożenia algorytmów z zakresu sztucznej inteligencji.

Słuchacze?

- Kierunek i moment studiów?
- Doświadczenie z optymalizacją ogólną?
- Doświadczenie z problemami optymalizacyjnymi? (TSP, VRP, BPP)?
- Doświadczenie z Javą i C++?

Zakres przedmiotu I

- Zadanie optymalizacji
Czyli co?
- Czy istnieje najlepszy algorytm optymalizacyjny?
O szukaniu igły w stogu siana i darmowych obiadach.
- Co zrobić, żeby się nie narobić?
O projektowaniu funkcji celu i metodach losowych.
- Czy 1 jest bliżej 17 czy 2?
O sąsiedztwie i przestrzeniach przeszukiwania.
- Czy więcej znaczy lepiej?
O metodach populacyjnych, na przykładzie metod ewolucyjnych.

Zakres przedmiotu II

- Czego możemy nauczyć się od mrówek, pszczół, ryb i ptaków?
Co wspólnego mają Powrót Batmana, film Władca Pierścieni i serial Gra o Tron?
O inteligencji rojowej.
- Czy algorytmy heurystyczne są nam potrzebne?
O eksploracji, eksploatacji, hiper-heurystykach i algorytmie memetycznym.
- Jak oceniać algorytmy?
O zbiorach benchmarkowych i opisie wyników.
- Jak zadowolić klienta?
O optymalizacji wielokryterialnej, odpornej i meta-optymalizacji.

Zakres przedmiotu III

- Jak odbierać laptopy i odpady?
Studium przypadku: Vehicle Routing Problem
- Jak ciąć drewno?
Studium przypadku: 2-D Packing and Cutting Problem

Zakres przedmiotu IV

- Monte Carlo i hill-climbing
- Simulated Annealing i Variable Neighbourhood Search
- Evolutionary Algorithm i Genetic Algorithm
- Particle Swarm Optimization i Ant Colony Optimization
- Differential Evolution

Zakres przedmiotu: projekt

- Rozwój podejścia *Generalized Self-Adapting Particle Swarm Optimization* (testowanie na benchmarkach COCO BBOB)
- Benchmarki CEC (2017)
- Konkurs BBComp

Umowa

- **Prowadzący** - przygotowuje najlepsze zajęcia jakie potrafi i pokazuje tropy którymi warto podążać
- **Studenci** - robią notatki i w trakcie zajęć podążają za tymi tropami
- **Prowadzący** - luźno podchodzi do oceniania laboratorium
- **Studenci** - rzetelnie robią zadania i kończą je w domu na następne laboratorium, żeby mogli kontynuować pracę
- **Razem** - szukamy najfajniejszych algorytmów optymalizacyjnych i staramy się zostawić coś po sobie (MIT/Expat License)

Umowa

- **Prowadzący** - przygotowuje najlepsze zajęcia jakie potrafi i pokazuje tropy którymi warto podążać
- **Studenci** - robią notatki i w trakcie zajęć podążają za tymi tropami
- **Prowadzący** - luźno podchodzi do oceniania laboratorium
- **Studenci** - rzetelnie robią zadania i kończą je w domu na następne laboratorium, żeby mogli kontynuować pracę
- **Razem** - szukamy najfajniejszych algorytmów optymalizacyjnych i staramy się zostawić coś po sobie (MIT/Expat License)

Umowa

- **Prowadzący** - przygotowuje najlepsze zajęcia jakie potrafi i pokazuje tropy którymi warto podążać
- **Studenci** - robią notatki i w trakcie zajęć podążają za tymi tropami
- **Prowadzący** - luźno podchodzi do oceniania laboratorium
- **Studenci** - rzetelnie robią zadania i kończą je w domu na następne laboratorium, żeby mogli kontynuować pracę
- **Razem** - szukamy najfajniejszych algorytmów optymalizacyjnych i staramy się zostawić coś po sobie (MIT/Expat License)

Umowa

- **Prowadzący** - przygotowuje najlepsze zajęcia jakie potrafi i pokazuje tropy którymi warto podążać
- **Studenci** - robią notatki i w trakcie zajęć podążają za tymi tropami
- **Prowadzący** - luźno podchodzi do oceniania laboratorium
- **Studenci** - rzetelnie robią zadania i kończą je w domu na następne laboratorium, żeby mogli kontynuować pracę
- **Razem** - szukamy najfajniejszych algorytmów optymalizacyjnych i staramy się zostawić coś po sobie (MIT/Expat License)

Umowa

- **Prowadzący** - przygotowuje najlepsze zajęcia jakie potrafi i pokazuje tropy którymi warto podążać
- **Studenci** - robią notatki i w trakcie zajęć podążają za tymi tropami
- **Prowadzący** - luźno podchodzi do oceniania laboratorium
- **Studenci** - rzetelnie robią zadania i kończą je w domu na następne laboratorium, żeby mogli kontynuować pracę
- **Razem** - szukamy najfajniejszych algorytmów optymalizacyjnych i staramy się zostawić coś po sobie (MIT/Expat License)

Umowa

- **Prowadzący** - przygotowuje najlepsze zajęcia jakie potrafi i pokazuje tropy którymi warto podążać
- **Studenci** - robią notatki i w trakcie zajęć podążają za tymi tropami
- **Prowadzący** - luźno podchodzi do oceniania laboratorium
- **Studenci** - rzetelnie robią zadania i kończą je w domu na następne laboratorium, żeby mogli kontynuować pracę
- **Razem** - szukamy najfajniejszych algorytmów optymalizacyjnych i staramy się zostawić coś po sobie (MIT/Expat License)

Zaliczenie

- Powstaje przy okazji pracy, a nie jest celem

Zaliczenie

- 60 punktów - laboratorium
- 40 punktów - projekt



Cel przedmiotu



= ?



Cel przedmiotu



≠





Cel przedmiotu



=



Literatura I

- 1 Shi, Y.; Eberhart, R.C. (1998). "A modified particle swarm optimizer". Proceedings of IEEE International Conference on Evolutionary Computation. pp. 69–73.
- 2 Storn, R.; Price, K. (1997). "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces". Journal of Global Optimization. 11: 341–359
- 3 Wolpert, D.H., Macready, W.G. (1997), "No Free Lunch Theorems for Optimization", IEEE Transactions on Evolutionary Computation 1, 67.
- 4 Gendreau, M.; Potvin, J-Y. (2010). "Handbook of Metaheuristics". Springer.

Literatura II

- 5 Panigrahi, B.K.; Shi, Y.; Lim, M. (2011), "Handbook of swarm intelligence: concepts, principles and applications". Springer.
- 6 Arabas J. (2004). „Wykłady z algorytmów ewolucyjnych”. WNT

Zadanie optymalizacji

- Szukamy parametrów modelu matematycznego pewnego układu(*), które zapewnią poprawę(**) działania tego układu
- Problem reprezentujemy jako funkcję jakości (przystosowania, celu)
- Poszukiwane parametry są argumentami tej funkcji
- $X^* = \{x^* : \forall x \in D f(x) \preceq f(x^*)\}$



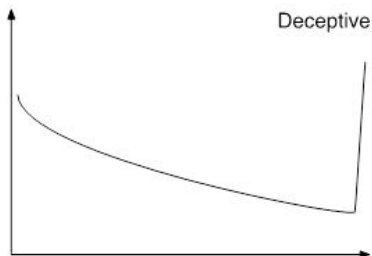
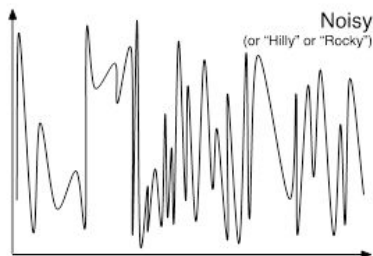
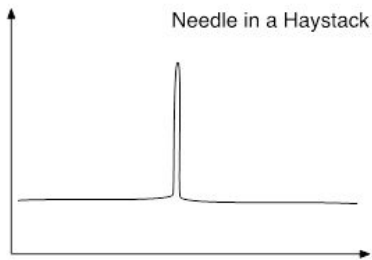
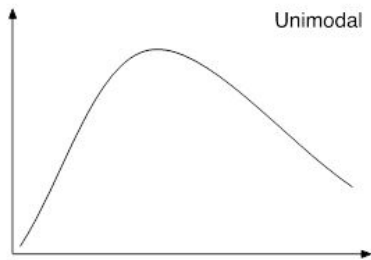
Poszukiwania trwają



<https://github.com/fcampelo/EC-Bestiary#introduction>



<http://slideplayer.com/slide/4579308/>





Zadanie

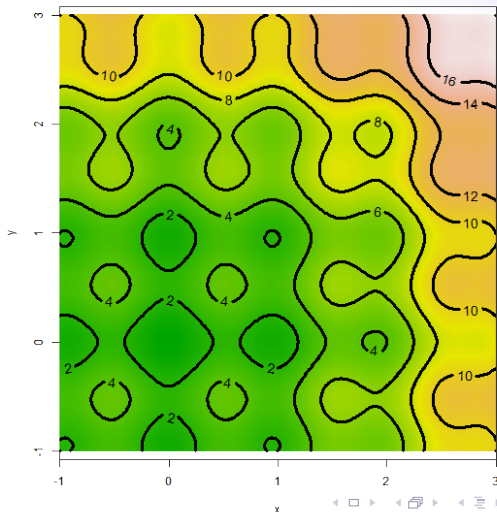
Znaleźć optimum funkcji $f : \mathbb{R}^n \rightarrow \mathbb{R}$,
np. $f(x) = n + \sum_{i=1}^n (x_i^2 - \cos(2\pi x_i))$

```
f <- function(x) {  
  if (!is.matrix(x)) {  
    x = matrix(x, ncol = length(x))  
  }  
  rowSums(x^2) - rowSums(cos(2*pi*x)) + ncol(x)  
}
```



Zadanie

Funkcja Rastrigina





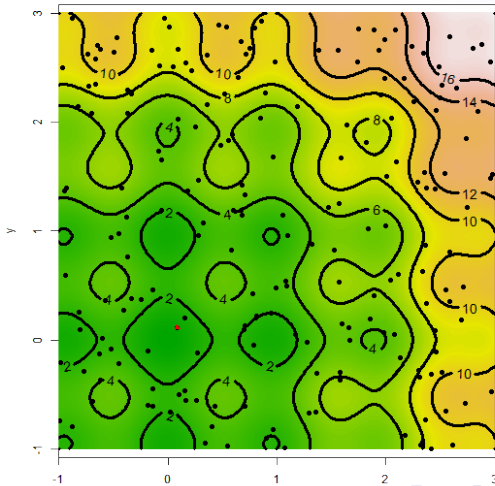
Rozwiązanie (dla leniwych)

```
samples.count = 200
sample = matrix(
  runif(
    dim*samples.count,
    min(x),max(x)
  ),
  ncol=dim,
  nrow=samples.count)
values = f(sample)
sample[which.min(values),]
```




Rozwiązanie (dla leniwych)

Funkcja Rastrigina





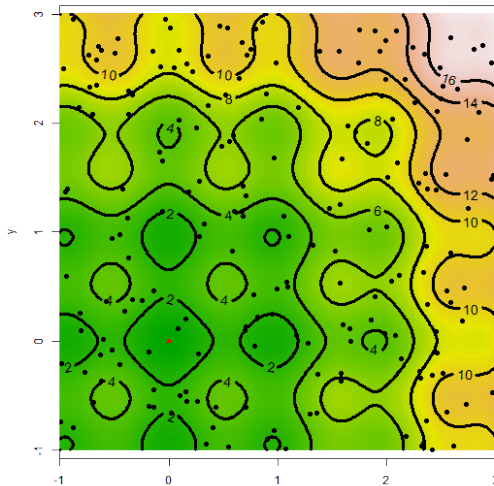
Rozwiązanie (dla lubiących gradienty)

```
res = optim(fn = f,  
           par = sample[which.min(values),],  
           method = "L-BFGS-B",  
           lower = min(x),  
           upper = max(x))  
res$par
```



Rozwiązanie (dla lubiących gradienty)

Funkcja Rastrigina



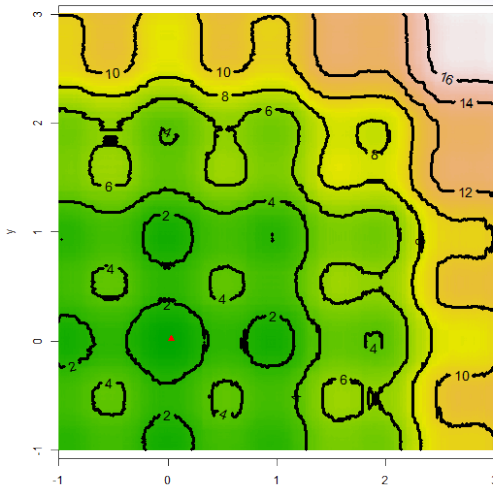


Rozwiązanie (dla statystyków)

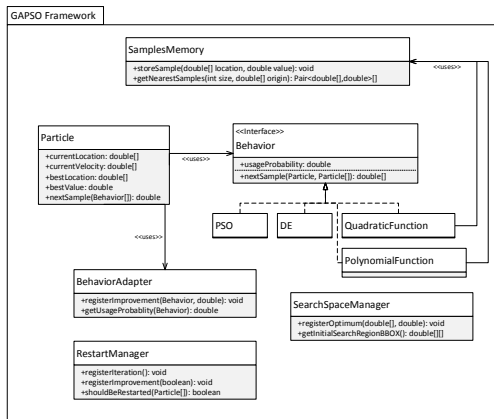
```
require(randomForest)
rf = randomForest(x = sample, y = values)
z = outer(X=x,
          Y=y,
          FUN=function(X1, X2)
            {predict(rf, data.frame(cbind(X1, X2)))})
```



Rozwiązanie (dla statystyków)



A gdyby połączyć te metody w jeden algorytm?



<https://bitbucket.org/pl-edu-pw-mini-optimization/basic-pso-de-hybrid/>

Praca domowa

- Zaimplementować algorytm *Hill Climbing* (przełączanie bitów) dla zdyskretyzowanej przestrzeni przeszukiwania
- Zaprezentować porównanie sąsiedztwa dla podejścia ciągłego i dyskretnego
- Porównać jakość obu podejść (parametrem dyskretnego HC powinien być poziom dyskretyzacji)

<https://imgur.com/gallery/iK0aV7D>

**YOU CAME TO THE WRONG
NEIGHBORHOOD**

