Symulacja wyżarzania

Metoda Monte Carlo

Maria Budziło, Marcin Kubański, Hiacynta Stypuła

25.01.2022

Politechnika Krakowska
Wydział Inżynierii
Materiałowej i Fizyki

Plan prezentacji

- Wstęp. Opis problemu
- Kryterium Metropolis
- Rozwiązanie problemu skrajnych temperatur
- Algorytm bazowy
- Rozwiązanie problemu. Implementacja kodu
- Porównanie z innymi rozwiązaniami

Wstęp. Opis problemu

Wyżarzanie jest procesem chłodzenia stopionej substancji. Celem wyżarzania jest skondensowanie materii w krystaliczne ciało stałe.

Wyżarzanie można potraktować jako proces optymalizacji.

$$exp(-E(r_i)/kT),$$

$E(r_i)$ - energia konfiguracji
k - stała Boltzmanna
T - temperatura

Zadaniem będzie osiągnięcie procedury symulacji wyżarzania.

Kryterium Metropolis

Technika iteracyjnego udoskonalania dla optymalizacji kombinatorycznej $\sim$ szybkie hartowanie stopionych metali

Podczas wyżarzania, nowa konfiguracja układu jest akceptowana na podstawie współczynnika prawdopodobieństwa Boltzmanna tej konfiguracji. To kryterium akceptacji nowego stanu systemu nazywane jest **kryterium Metropolis**.

Rozwiązanie problemu skrajnych temperatur

Dobrym pomysłem na uniknięcie wielu problemów związanych ze znalezieniem rozwiązania jest zastosowanie zbioru zamiast pojedynczego rozwiązania.

Poprzez uwrażliwienie szybkości zmiany temperatury na jakość rozwiązania, mechanizm wyżarzania może być sprzężony z jakością aktualnego rozwiązania.

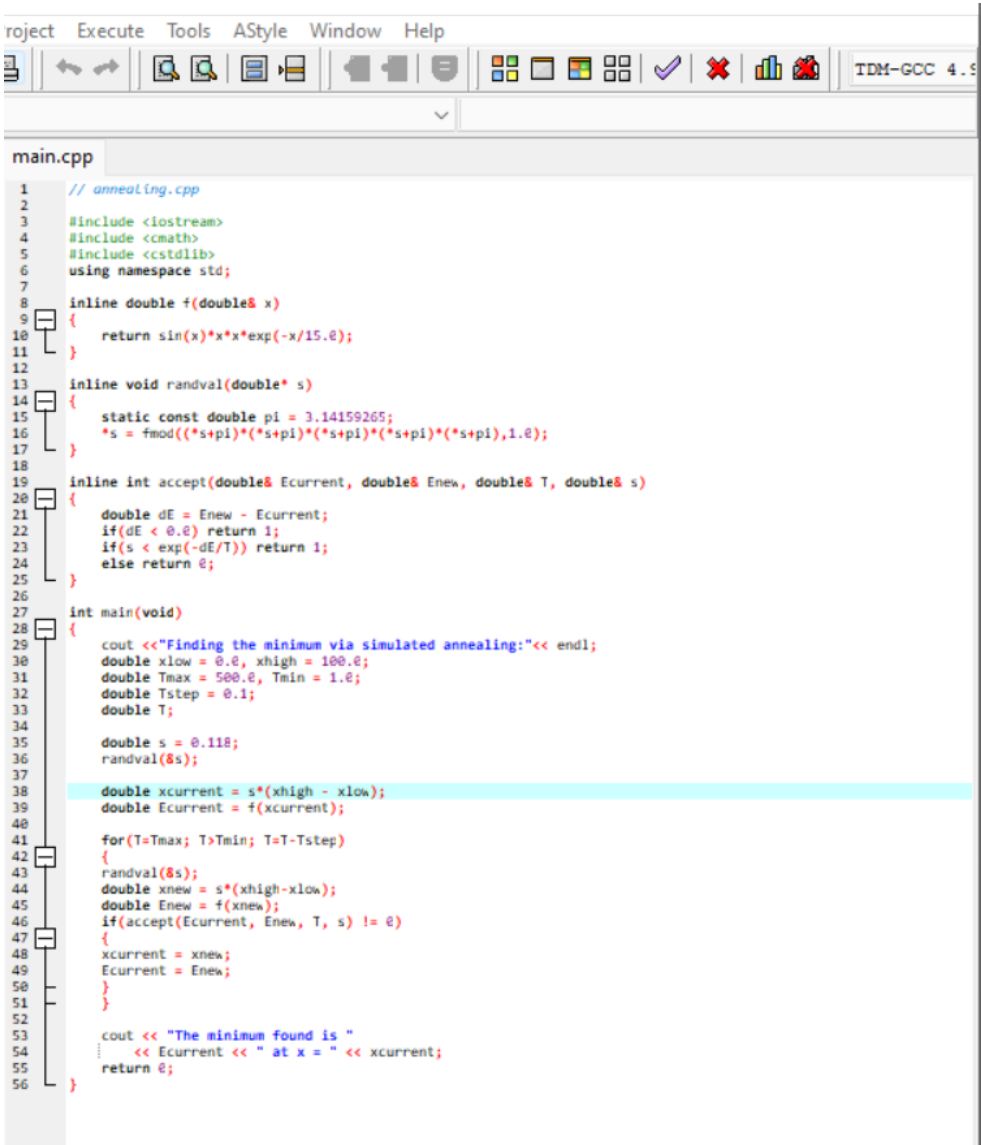
Algorytm bazowy

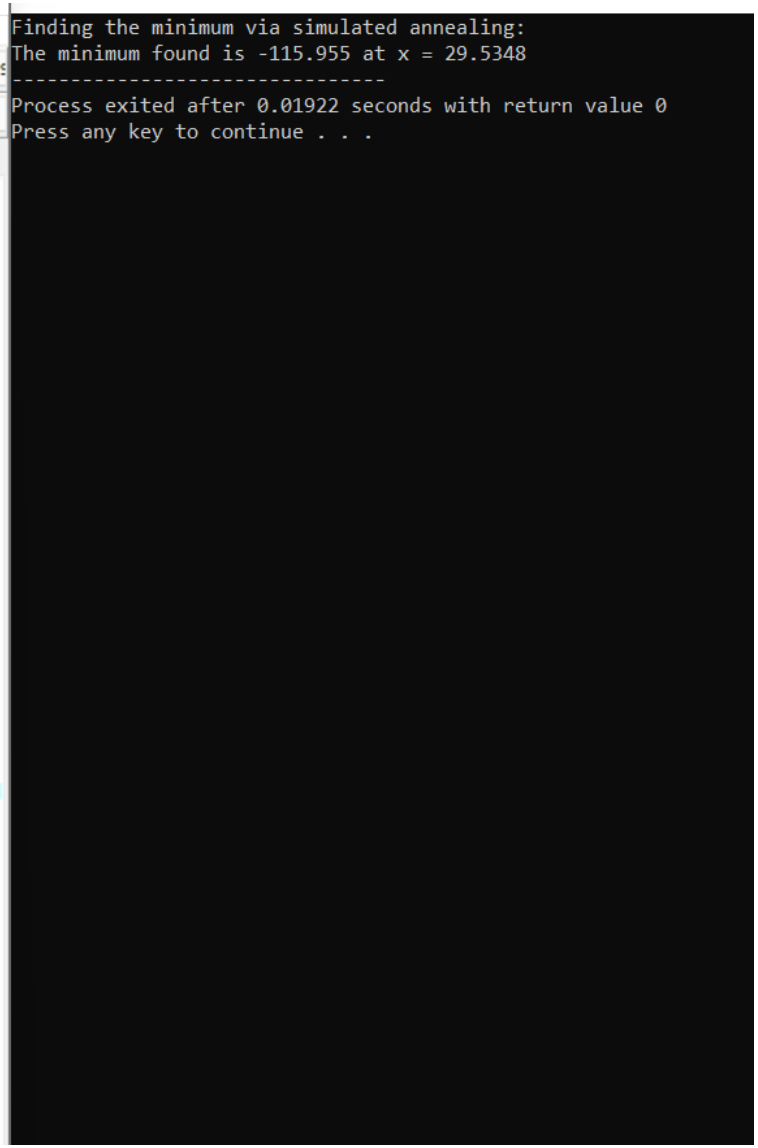
Symulacja wyżarzania polega w istocie na powtarzaniu procedury Metropolis dla różnych temperatur.

Procedura symulowanego wyżarzania:

początek
t <- 0
ustawiamy temperaturę T
wybieramy losowo aktualny ciąg v_c
oceniamy sprawność v_c
powtarzamy
powtarzamy
wybieramy nowy ciąg v_n
w pobliżu v_c
poprzez odwrócenie pojedynczego bitu v_c
jeśli f(v_c)< f(v_n)
wtedy v_c <- v_n
inaczej, jeśli losowo[0,1] < exp((f(v_n) - f(v_c)/T)
wtedy v_c <- v_n
dopóki (warunek zakończenia)
T <- g(T,t)
t <- t+ 1
aż do (kryterium zatrzymania)
koniec

Rozwiązanie problemu. Implementacja kodu

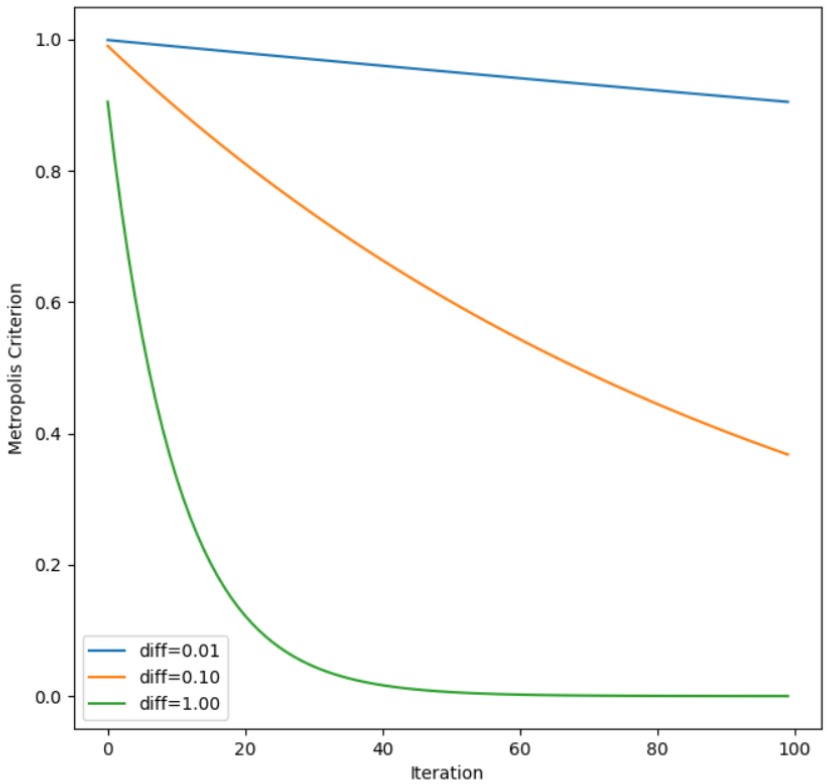




Porównanie z innymi rozwiązaniami

wyz.py

```python
# explore metropolis acceptance criterion for simulated annealing
from math import exp
from matplotlib import pyplot
# total iterations of algorithm
iterations = 100
# initial temperature
initial_temp = 10
# array of iterations from 0 to iterations - 1
iterations = [i for i in range(iterations)]
# temperatures for each iterations
temperatures = [initial_temp/float(i + 1) for i in iterations]
# metropolis acceptance criterion
differences = [0.01, 0.1, 1.0]
for d in differences:
    metropolis = [exp(-d/t) for t in temperatures]
    # plot iterations vs metropolis
    label = 'diff=%.2f' % d
    pyplot.plot(iterations, metropolis, label=label)
# inalize plot
pyplot.xlabel('Iteration')
pyplot.ylabel('Metropolis Criterion')
pyplot.legend()
pyplot.show()
```



Bibliografia


Materiały udostępnione w pliku „GeneticAlgorithms.pdf”


https://machinelearningmastery.com/simulated-annealing-from-scratch-in-python/

Dziękujemy za uwagę!