METODY NUMERYCZNE – LABORATORIUM NR 10

Poszukiwanie minimum wartości funkcji metodą symulowanego wyżarzania

1. Wstęp

Celem zajęć będzie znalezienie minimum funkcji

$$f(x,y) = \sin(x)\sin(y) - \exp\left(-\left(x+\frac{\pi}{2}\right)^2 - \left(y-\frac{\pi}{2}\right)^2\right) \tag{1}$$

przy pomocy metody **symulowanego wyżarzania ze zmienną temperaturą** T zaprogramowanej dla N wędrowców – każdy wędrowiec ma w danej chwili współrzędne (x_i, y_i) (indeks i oznacza numer wędrowca, $i = 0, 1, \ldots, N-1$). Aktualne współrzędne wszystkich wędrowców trzeba przechowywać w tablicach.

Symulowane wyżarzanie to algorytm typu **Monte Carlo**: polega na wielokrotnym losowaniu przesunięć położeń wędrowców. Jeśli w nowym położeniu wędrowca funkcja f przyjmuje mniejszą wartość, to nadpisujemy stare położenie wędrowca nowym (mówimy o akceptacji nowego położenia). W przeciwnym wypadku nowe położenie akceptujemy tylko z pewnym prawdopodobieństwem, zadanym przez **rozkład Boltzmanna**: przy wysokiej temperaturze jest duże prawdopodobieństwo akceptacji "gorszego" położenia, przy niskiej – niewielkie. Schemat algorytmu:

1. Wybieramy **współrzędne początkowe** $\left(x^{(0)}, y^{(0)}\right)$ i przypisujemy je wszystkim wędrowcom.

2. `for` $(i_\mathrm{T} = 0;\ i_\mathrm{T} <= 20;\ i_\mathrm{T} = i_\mathrm{T} + 1)$ — *// Pętla zewnętrzna* ***obniżająca temperaturę****.*

3. $T = \dfrac{10}{2^{i_\mathrm{T}}}$ — *// Ustalenie* ***aktualnej temperatury****.*

4. `for` $(k = 0;\ k < 100;\ k = k+1)$ — *// 100 kroków...*

5. `for` $(i = 0;\ i < N;\ i = i+1)$ — *// ...dla każdego z wędrowców.*

6. $\Delta x_i =$ `d_rand(-1, 1)` — *//* ***Losujemy przesunięcia*** $\Delta x_i, \Delta y_i$ *z przedziału* $[-1, 1]$,

7. $\Delta y_i =$ `d_rand(-1, 1)` — *// uważając, żeby wędrowcy* ***nie wyszli*** *poza* $[x_\mathrm{min}, x_\mathrm{max}] \times [y_\mathrm{min}, y_\mathrm{max}]$.

8. `if` $\Big(f\left(x_i + \Delta x_i, y_i + \Delta y_i\right) < f\left(x_i, y_i\right)\Big)$ — *// Jeżeli nowe położenie jest lepsze...*

9. $x_i = x_i + \Delta x_i$ — *// ...to je akceptujemy.*

10. $y_i = y_i + \Delta y_i$

11. `else if` $\left(\texttt{d\_rand(0, 1)} < \exp\left(-\dfrac{f\left(x_i + \Delta x_i, y_i + \Delta y_i\right) - f\left(x_i, y_i\right)}{T}\right)\right)$

12. $x_i = x_i + \Delta x_i$ — *// Gorsze położenie akceptujemy z prawdopodobieństwem*

13. $y_i = y_i + \Delta y_i$ — *// danym przez rozkład Boltzmanna.*

14. Po zakończeniu algorytmu wyszukujemy wędrowca, dla którego wartość funkcji f jest najmniejsza – jego położenie uznajemy za znalezione **minimum**.

Uwaga: proszę się upewnić, że wędrowcy nie wychodzą poza interesujący nas obszar $[x_\mathrm{min}, x_\mathrm{max}] \times [y_\mathrm{min}, y_\mathrm{max}]$. Np. jeśli nowa współrzędna $(x_i + \Delta x_i)$ byłaby większa od x_max, to można "uciąć" Δx_i do największej możliwej wartości (lub wylosować na nowo, "do skutku") i dopiero przejść do kroku **8.**

Powyższy pseudokod zakłada, że funkcja `d_rand` zwraca pseudolosową liczbę zmiennoprzecinkową z przedziału podanego przez argumenty. Przykładowa implementacja:

```
double d_rand(const double min, const double max)
{
  double r = (double)rand() / RAND_MAX; // Przedzial [0, 1]
  r = r * (max - min) + min;            // Przeskalowanie do [min, max]
  return r;
}
```

2. Zadania do wykonania

Naszym zadaniem jest zaprogramowanie powyższego algorytmu dla $N = 200$ wędrowców, startujących od położenia początkowego $\left(x^{(0)}, y^{(0)}\right) = (5, 5)$. Minimum funkcji szukamy w obszarze $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] = [-10, 10] \times [-10, 10]$. Dla każdego wędrowca należy wykonać 100 kroków błądzenia dla każdej temperatury. Należy wykonać następujące rysunki:

2.1. Na wykresie konturowym funkcji (1) nanieść aktualne położenia wędrowców po zakończeniu błądzenia dla trzech przypadków temperatur: $i_{\mathrm{T}} = 0$ (wówczas $T = 10$), $i_{\mathrm{T}} = 7$ ($T = 0.078125$) oraz po zakończeniu algorytmu, tj. dla $i_{\mathrm{T}} = 20$ ($T \approx 0.0000095$). *Wypisywanie powinno zostać zaimplementowane po zakończeniu pętli po k.*

2.2. Dla jednego z wędrowców (np. dla pierwszego) wypisać i narysować wartości funkcji $f\left(x_i, y_i\right)$ dla wszystkich jego położeń w trakcie działania całego algorytmu.

W sprawozdaniu proszę zamieścić wykresy oraz odpowiedzieć na pytania:

- Czy znalezione minimum jest globalne?
- Jaki wpływ na działanie algorytmu mają: temperatura, liczba wędrowców?
- Czy uruchomienie algorytmu dla jednego wędrowca ($N = 1$) byłoby skuteczne?
- Czy obniżanie temperatury jest konieczne?