Heat Flow 3: Annealing - Proces wyżarzania

Krzysztof Konieczny, Wojciech Kura

styczeń 2021,
Politechnika Krakowska im. Tadeusza Kościuszki w Krakowie

Spis treści

1 Wstęp

Wyżarzanie to proces chłodzenia stopionej substancji w celu skroplenia materii w krystaliczną substancję stałą. Można go potraktować jako proces optymalizacji. Podczas wyżarzania konfiguracja systemu jest określona przez zbiór pozycji atomowych r_i. Jest ona ważona współczynnikiem prawdopodobieństwa Boltzmanna, a przedstawia się to następującym wzorem:

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

gdzie $\mathbf{E}(r_i)$ to energia konfiguracji, **k** to stała Boltzmanna, a **T** to temperatura. Podczas poddawaniu substancji procesowi wyżarzania, każda temperatura jest utrzymywana przez odpowiednio długi czas, aż do osiągnięcia równowagi termicznej.

2 Kryterium Metropolii, przebieg wyżarzania

Powtarzalną technikę doskonalenia optymalizacji kombinatorycznej możemy porównać z szybkim hartowaniem stopionych metali. Podczas szybkiego hartowania, energia jest błyskawicznie usuwana z układu poprzez kontakt z zimną, masywną substancją. Takie drastyczne ochładzanie powoduje metastabilne stany systemu, np. w metalurgii w wyniku szybkiego chłodzenia otrzymuje się raczej szklistą substancję niż krystaliczną, stałą. Analogia między powtarzalnym ulepszaniem a szybkim chłodzeniem metali przyjmuję tylko te konfiguracje systemu, które zmniejszają funkcję sprawności. W procesie wyżarzania (czyli powolnego chłodzenia) nowa konfiguracja systemu, która nie poprawia funkcji kosztu, jest akceptowana w oparciu o czynnik prawdopodobieństwa Boltzmanna. Przyjmując nowy stan systemu uzyskujemy tzw. **"kryterium Metropolii"**.

Jeśli temperatura początkowa jest zbyt niska, proces zostaje szybko zatrzymany i znajdujemy tylko lokalne optimum, lecz gdy jest zbyt wysoka, proces ten przebiega bardzo wolno. Do wyszukiwania używamy tylko jednego rozwiązania, co zwiększa prawdopodobieństwo utknięcia rozwiązania w lokalnym optimum. Zmiana temperatury oparta jest na zewnętrznej procedurze niezwiązanej z aktualną jakością roztworu, czyli szybkością zmian temperatury, niezależną od jakości roztworu. Problemy te rozwiązujemy, stosując populację zamiast tylko jednego rozwiązania. Mechanizm wyżarzania można również powiązać z jakością obecnego roztworu, zmieniając częstotliwość szybkości zmian temperatury na jakość roztworu.

3 Kod i opis programu

Poniżej przedstawiamy kod programu, który pokazuję procedurę wyżarzania. Polega on zasadniczo na powtarzaniu kryterium Metropolii dla różnych temperatur, które są stopniowo obniżane przy każdej iteracji algorytmu. Użyliśmy w tym celu procesu wyżarzania do znalezienia minimum funkcji:

$$f(x) = x^2 \sin(x) \exp(-x/15.0) \tag{1}$$

```
#include <iostream>
#include <cmath>
#include <cstdlib>
using namespace std;

inline double f(double& x)
{
    return sin(x)*x*x*exp(-x/15.0); //funkcja goes
        there
}

inline void randval(double*s)
{
    static const double pi = 3.14159265;
    *s=fmod((*s+pi)*(*s+pi)*(*s+pi)*(*s+pi)*(*s+pi)
        ,1.0);
}
```

```
inline int accept(double& Eobecne,double& Enowe, double
   & T, double& s)
{
    double dE = Enowe - Eobecne;
    if(dE < 0.0) return 1;
    if(s < exp(-dE/T)) return 1;
    else return 0;
}
int main(void)
{
    cout << "Szukanie minimum poprzez symulowane
       wyzarzanie:" << endl;
    double xlow = 0.0, xhigh = 100.0;
    double Tmax = 500.0, Tmin = 1.0;
    double Tkrok = 0.1;
    double T;
    double s = 0.118;  //seed
    randval(&s);

    double xobecne = s*(xhigh - xlow);
    double Eobecne = f(xobecne);

    for(T=Tmax; T>Tmin; T=T-Tkrok)
    {
        randval(&s);
        double xnowe = s*(xhigh-xlow);
        double Enowe = f(xnowe);
        if(accept(Eobecne,Enowe,T,s) != 0)
        {
            xobecne = xnowe;
            Eobecne = Enowe;
        }
    }

    cout << "Znalezione minimum to " << Eobecne << " w
       x = " << xobecne;
    return 0;
}
```

4 Wyniki działania programu

Program po uruchomieniu zwraca w konsoli wyniki dla funkcji podanej w kodzie. W pierwszym przypadku jest to funkcja (1)

```
PS C:\Users\User\Desktop\nauka\V semestr\modelproj3\proj> .\proj3.exe
Szukanie minimum poprzez symulowane wyzarzanie:
Znalezione minimum to -115.955 w x = 29.5348
```

Uruchomiliśmy program również dla innych funkcji takich jak:

$$f(x) = x^2 \cos(x) \exp(2x/37.0) \tag{2}$$

Z wynikiem:

```
PS C:\Users\User\Desktop\nauka\V semestr\modelproj3\proj> .\proj3.exe
Szukanie minimum poprzez symulowane wyzarzanie:
Znalezione minimum to -1.83857e+06 w x = 97.4614
```

Oraz:

$$f(x) = x^2 \sin(x) \exp(-3x/22.0) \tag{3}$$

Z czego otrzymaliśmy poniższy wynik:

```
PS C:\Users\User\Desktop\nauka\V semestr\modelproj3\proj> .\proj3.exe
Szukanie minimum poprzez symulowane wyzarzanie:
Znalezione minimum to -27.0208 w x = 11.0414
```

5 Bibliografia

- *R.H. Landau et al.,* "Computational Physics: Problem Solving with Computers, Wiley