

Symulowane wyżarzanie

1. Charakterystyka

Za początek algorytmów symulowanego wyżarzania (Simulated Annealing – SA) przyjmuje się pracę Kirkpatrick, Gelatt i Vecchi z 1983 roku [261]. Inspiracją do powstania tych algorytmów był proces nadzorowanego wyżarzania polegający na utrzymywaniu określonych temperatur przez określone okresy czasu. Jest on przeprowadzany w celu uzyskania pożądanych własności materiału, zasadniczo redukcji defektów i naprężeń. W przypadku metali chodzi także o uzyskanie sieci krystalicznej o możliwie najniższej energii i możliwie dużych kryształach metalu. Proces ten przeprowadza się także w przypadku innych materiałów nie tworzących sieci krystalicznej – czyli materiałów amorficznych np. szkło.

Algorytm symulowanego wyżarzania rozpoczyna się od losowego wygenerowania rozwiązania postawionego problemu (rysunek 3.14), które jednocześnie staje się rozwiązaniem bieżącym. Następnie tworzy się rozwiązanie pochodne w stosunku do rozwiązania bieżącego. W zależności od problemu optymalizacyjnego i przyjętej konstrukcji algorytmu, zwykle sprowadza się ono do wykonania drobnej losowej modyfikacji bieżącego rozwiązania. Kolejnym krokiem jest sprawdzenie, czy nowe rozwiązanie jest lepsze od bieżącego - jeżeli tak jest, to nowe rozwiązanie staje się rozwiązaniem bieżącym. W przeciwnym przypadku, nowe rozwiązanie może stać się rozwiązaniem bieżącym jeżeli wyznaczona wartość prawdopodobieństwa p jest większa od losowej liczby r .

$$p = e^{-\frac{\Delta E}{k_B T}} \quad (3.1)$$

Wartość p wyznaczana według formuły (3.1) zależy od:

ΔE – zmiana poziomu energii, którą wprowadza nowe rozwiązanie,

k_B – stała Boltzmanna,

T – bieżąca temperatura.

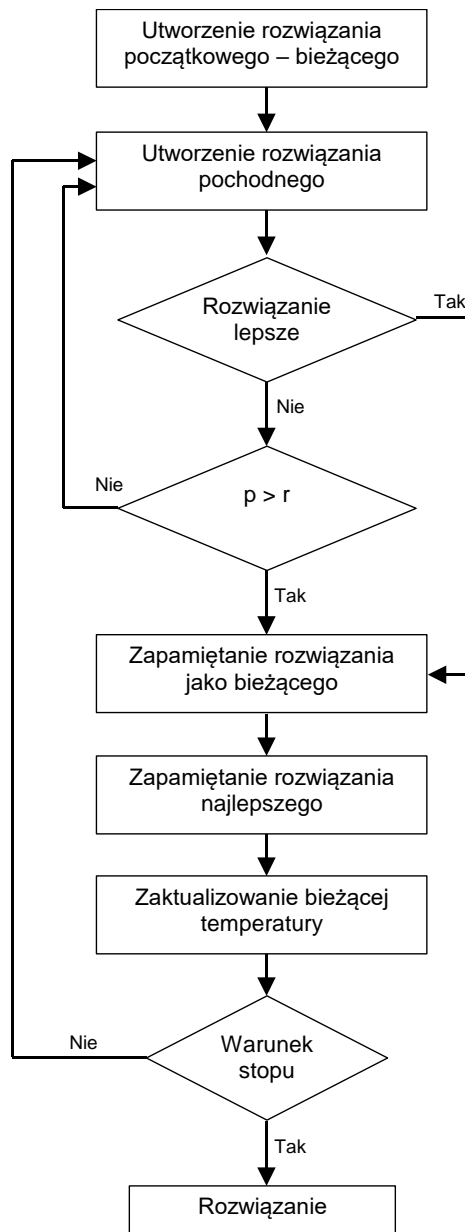
Przy czym:

$$\Delta E = \gamma \cdot \Delta f$$

gdzie:

Δf – zmiana wartości funkcji celu, dla zadań maksymalizowanych różnica między wartością nową a bieżącą,
 γ – stała.

Zwykle korzysta się z uproszczonej zależności przyjmując k_B i γ równe jedności.



Rysunek 3.14 Ogólna struktura algorytmu symulowanego wyżarzania

Po zapamiętaniu najlepszego do tej pory osiągniętego rozwiązania, aktualizuje się bieżącą temperaturę układu. Istnieje wiele sposobów jej aktualizacji – zmniejszania. Najczęściej stosuje się wychładzanie o określoną wartość bezwzględną lub względną. Ostatnim

elementem tego algorytmu jest sprawdzenie warunku zatrzymania. Na ogół jest to osiągnięcie założonej temperatury minimalnej przez układ lub przeprowadzenie określonej liczby iteracji. Algorytm symulowanego wyżarzania jest uważany, przy przyjęciu określonych parametrów jego działania, za algorytm optymalizacji globalnej. Kluczowe są tu dwa czynniki. Pierwszym jest wystarczająco długie schładzanie układu. Drugim sposób generowania rozwiązań pochodnych.

Mimo swojej prostoty, algorytm tego rodzaju po pierwszej fazie swojej popularności został wyparty przez algorytmy ewolucyjne.

Zastosowania

Pierwsze zastosowania symulowanego wyżarzania w elektroenergetyce pojawiły się na przełomie lat 80-tych i 90-tych XX wieku jako jedna z pierwszych metod z rodziny metaheurystyk, która pozwalała na optymalizowanie różnych zadań bez bezpośredniej wiedzy o kształcie funkcji celu. Szczególnie duże nadzieje pokładano w możliwości optymalizowania zadań o zmiennych dyskretnych i kombinatorycznych. Metoda nie zyskała jednak dużej popularności, o czym świadczy niewielka liczba publikacji, szacowana na dwa rzędy wielkości mniejszą niż publikacji poświęconych algorytmom ewolucyjnym. Te ostatnie bowiem dość szybko wyparły symulowane wyżarzanie, mimo swojej większej złożoności. Obecnie metody oparte na symulowanym wyżarzaniu służą albo jako metoda referencyjna dla nowszych metod, albo jako składnik rozwiązania hybrydowego.

Prognozowanie

Symulowane wyżarzanie posiada bardzo niewielką liczbę aplikacji w prognozowaniu. Ich zastosowania w prognozowaniu krótkoterminowym sprowadzają się do optymalizacji parametrów pewnego modelu predykcyjnego – podobnie jak to ma miejsce w przypadku innych typowych metod optymalizacyjnych (np. PSO i AE).

Optymalizacja

Optymalizacja stanowi największą grupę zastosowań symulowanego wyżarzania, co jest zgodne z naturą tej metody. Do typowych aplikacji należą:

1. Ekonomiczny rozdział obciążeń
2. Optymalizacja rozptywu mocy
3. Optymalizacja lokalizacji kompensatorów mocy biernej
4. Optymalizacja lokalizacji urządzeń łączeniowych w sieci rozdzielczej

5. Optymalizacja lokalizacji generacji rozproszonej w sieci
6. Optymalizacja struktury sieci
7. Optymalizacja konfiguracji sieci
8. Optymalizacja konstrukcji generatora
9. Optymalizacja konstrukcji transformatora

Identyfikacja

Wykorzystanie symulowanego wyżarzania w tym obszarze, mimo, że posiada niewielką reprezentację, charakteryzuje się bardzo szerokim spektrum aplikacji. Ich wspólnym mianownikiem jest optymalizowanie pewnego zadania lub podzadania za pomocą SA:

1. Optymalizacja lokalizacji urządzeń pomiarowych w sieci
2. Podział sieci na potrzeby estymacji stanu
3. Lokalizacja zwarć w sieci
4. Ocena stabilności systemu elektroenergetycznego
5. Estymacja kosztu energii w sieci
6. Monitorowanie sygnałów pomiarowych

Sterowanie

Liczba zastosowań SA w zakresie zagadnień sterowania jest dość symboliczna. Obejmuje ona typowe dla metod optymalizacyjnych zastosowania symulowanego wyżarzania do optymalizacji parametrów stabilizatorów systemowych i sterowników kompensatorów szeregowych.

2. Funkcje testowe

- Funkcja Ackley'a

$$f(x) = -20 \exp \left[-\frac{1}{5} \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right] - \exp \left[\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right] + 20 + e$$

gdzie: $n = 1, 2, \dots$; and $-32.768 \leq x_i \leq 32.768$ for $i = 1, 2, \dots, n$. Minimum globalne funkcji $f_* = 0$ osiągnane jest dla $x_* = (0, 0, \dots, 0)$.

- Funkcja Rosenbrocka:

$$f(x) = \sum_{i=1}^{n-1} \left[(x_i - 1)^2 + 100(x_{i+1} - x_i^2)^2 \right]$$

gdzie: $n = 1, 2, \dots$; and $-30.0 \leq x_i \leq 30.0$ for $i = 1, 2, \dots, n$. Minimum globalne funkcji $f_* = 0$ osiągnane jest dla $x_* = (1, 1, \dots, 1)$.

- Funkcja DeJonga:

$$f(x) = \sum_{i=1}^n x_i^2$$

gdzie: $n = 1, 2, \dots$; oraz $-100.0 \leq x_i \leq 100.0$ dla $i = 1, 2, \dots, n$. Minimum globalne funkcji $f_* = 0$ osiągnane jest dla $x_* = (0, 0, \dots, 0)$.

- Funkcja Griewanka

$$f(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (13)$$

gdzie: $n = 1, 2, \dots$; oraz $-600.0 \leq x_i \leq 600.0$ dla $i = 1, 2, \dots, n$. Minimum globalne funkcji $f_* = 0$ osiągnane jest dla $x_* = (0, 0, \dots, 0)$.

- Funkcja Rastrigina :

$$f(x) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)] \text{ gdzie: } n = 1, 2, \dots; \text{ oraz } -5.12 \leq x_i \leq 5.12 \text{ for } i$$

$= 1, 2, \dots, n$. Minimum globalne funkcji $f_* = 0$ osiągnane jest dla $x_* = (0, 0, \dots, 0)$.

- Funkcja Schwefela:

$$f(x) = -\sum_{i=1}^n x_i \sin(\sqrt{|x_i|})$$

gdzie: $n = 1, 2, \dots$; oraz $-500 \leq x_i \leq 500$ dla $i = 1, 2, \dots, n$. Minimum globalne funkcji $f_* \approx -418.9829 \cdot n$ osiągnane jest dla $x_* = (420.9687, 420.9687, \dots, 420.9687)$.

W celu zapewnienia dodatniości funkcji dla wszystkich numerów wymiarów oraz zmiennych, następująca modyfikacja funkcji powinna podlegać zastosowaniu.

$$f(x) = -\frac{1}{n} \sum_{i=1}^n x_i \sin(\sqrt{|x_i|}) + 500$$

Minimum globalne funkcji $f_* \approx 81.0171$ występuje w tym samym punkcie, co dla funkcji oryginalnej, tj. $x_* = (420.9687, 420.9687, \dots, 420.9687)$.

Celem optymalizacji każdej funkcji jest jej minimalizacja. Algorytmy były jednak budowane tak, aby maksymalizować funkcję przystosowania. Konieczne było więc przejście między funkcją celu a przystosowania, dokonywane docelowo za pomocą:

$$F_{fitness} = \frac{1}{F_{obj} + 1}$$

3. Przebieg ćwiczenia

- Oprócz funkcji domyślnej zespół wybiera 3 inne funkcje testowe
- W pliku prototype.goi należy ustawić odpowiednią dla danej funkcji testowej liczbę zmiennych oraz zakresy zmiennych wejściowych
- Wybór kombinacji parametrów wejściowych w pliku auto_SA.bxt, umożliwiających zbadanie wpływu każdej zmiennej na efekty symulacji. Kombinacje pozostają takie same dla każdej funkcji testowej.
- Porównanie wyników na podstawie plików report_SA_auto.txt, w tym ocena jakości rozwiązania znalezionej oraz analiza zbieżności