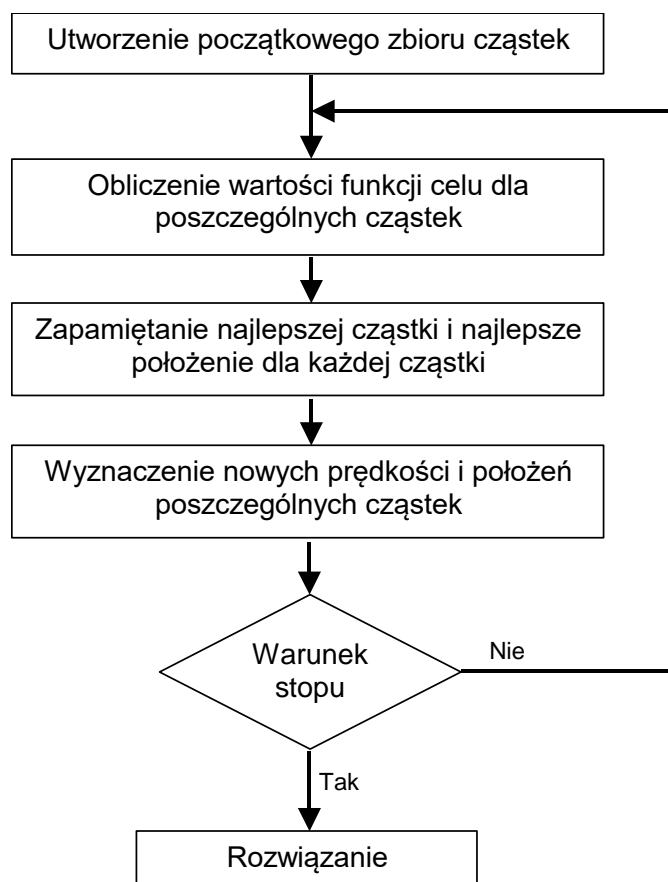


1. Optymalizacja rojem cząstek

Charakterystyka

Algorytmy optymalizacji rojem cząstek (Particle Swarm Optimization (PSO)) naśladują w swoim działaniu zachowanie stada żywych organizmów jako całości oraz poszczególnych jego członków poruszających się wspólnie w pewnym środowisku. Głównym aspektem naśladowanym przez PSO jest dynamika tego ruchu, w tym sposób przemieszczania w poszukiwaniu pokarmu i sposób unikania drapieżników. Ciekawa jest też kwestia propagacji informacji w stadzie. Badania w tym obszarze były prowadzone przez biologów przez wiele dekad. Pierwsze modele tych zjawisk pojawiły się w latach 80-tych XX wieku przede wszystkim na potrzeby grafiki komputerowej. W następnej dekadzie zjawiska te analizowano w kontekście rozwijającej się koncepcji sztucznego życia. Kluczową dla rozwoju PSO jako metody obliczeniowej była publikacja Kennedyego i Eberharta z 1995 roku, w której autorzy zaproponowali algorytm do optymalizacji funkcji nieliniowych.



Rysunek 3.15 Ogólna postać algorytmu PSO

Podstawową postać algorytmu PSO przedstawiono na rysunku 3.15. Algorytm ten rozpoczyna się od utworzenia początkowego zbioru cząstek. Dla każdej cząstki określa się, najczęściej w sposób losowy, położenie w wielowymiarowej przestrzeni rozwiązań oraz prędkość w każdym z wymiarów. Następnie oblicza się wartość funkcji celu dla każdej z cząstek. Kolejnym krokiem jest zapamiętanie najlepszego, do tej pory osiągniętego, położenia dla każdej z cząstek oraz dla wszystkich cząstek sumarycznie. Informacje te są potrzebne na etapie wyznaczania nowych prędkości i położen poszczególnych cząstek, który to proces w podstawowej, historycznej już postaci przebiegał według formuł (3.2) i (3.3). Wyznaczenie nowych prędkości i położen jest rozbite na poszczególne współrzędne cząstki. Wyznaczanie współrzędnej x_k^{t+1} cząstki w k-tym wymiarze, w kolejnym kroku (t+1) przeprowadza się najpierw obliczając prędkość v_k^{t+1} .

$$v_k^{t+1} = c_0 r_0^t v_k^t + c_1 r_1^t (y_k^t - x_k^t) + c_2 r_2^t (y_k^{p^*t} - x_k^t) \quad (3.2)$$

a następnie obliczając właściwą współrzędną w k-tym wymiarze:

$$x_k^{t+1} = x_k^t + v_k^{t+1} \quad (3.3)$$

przy czym:

r_0^t, r_1^t, r_2^t - to liczby losowe z przedziału (0;1),

v_k^t - prędkość cząstki w poprzednim kroku w k-tym wymiarze,

x_k^t - współrzędna cząstki w poprzednim kroku w k-tym wymiarze,

y_k^t - współrzędna cząstki reprezentująca jej najlepsze, do tej pory znalezione położenie,

$y_k^{p^*t}$ - współrzędna najlepszego znalezione do tej pory położenia (ze wszystkich cząstek we wszystkich iteracjach),

c_0 - współczynnik wagowy określający wpływ własnej prędkości cząstki na jej prędkość w kolejnej iteracji,

- c_1 - współczynnik wagowy określający wpływ najlepszego położenia danej cząstki na jej prędkość w kolejnej iteracji,
- c_2 - współczynnik wagowy określający wpływ najlepszego znalezionej do tej pory położenia wśród wszystkich cząstek populacji na prędkość danej cząstki w kolejnej iteracji,

Prędkość cząstki, przy założeniu długości czasu trwania jednej iteracji równej jedności, określa w istocie różnicę (wektor) w położeniu początkowym i końcowym.

Ostatnim etapem jest sprawdzenie warunku zatrzymania. Tak jak w innych metodach optymalizacyjnych, może to być osiągnięcie założonego poziomu błędu lub wykonanie określonej liczby iteracji algorytmu.

W czasie rozwoju algorytmów PSO powstało wiele różnych ich modyfikacji. Jedną z istotniejszych była zaproponowana przez Clerca i Kennedyego w 2002 roku nowa postać formuł do wyznaczania prędkości cząstek, która czyniła rój cząstek bardziej stabilnym. Obecnie istnieje już ponad 100 różnych wersji algorytmu PSO jako samodzielnej metody optymalizacyjnej i jako składnika układów hybrydowych. Najbardziej efektywne wydają się zastosowania PSO do optymalizacji parametrów działania sztucznych sieci neuronowych lub systemów rozmytych.

Zastosowania

Algorytmy optymalizacji rojem cząstek (PSO) są stosunkowo młodą grupą metod. Pierwsze aplikacje w elektroenergetyce pojawiły się na przełomie wieków. Liczba prac naukowych poświęconych tej tematyce rośnie od początku bardzo szybko i osiągnęła wielkość o rząd mniejszą od liczby prac poświęconych algorytmom ewolucyjnym. Algorytmy PSO są metodą podobną do algorytmów ewolucyjnych, podobnie do nich są metodą odporną i nie wymagają różniczkowalności czy ciągłości funkcji kryterialnej. Są także metodą łatwą w implementacji. Niestety nie są metodą tak uniwersalną, jak algorytmy ewolucyjne – zastosowanie ich do rozwiązywania zadań kombinatorycznych czy dyskretnych jest co najmniej trudne i nieoczywiste. Natomiast w rozwiązywaniu pozostałych zadań najczęściej charakteryzują się dużo większą efektywnością niż algorytmy ewolucyjne.

Podobnie jak algorytmy ewolucyjne, algorytmy PSO są z natury rzeczy metodą optymalizacyjną i mimo tego, że stosowane w różnych problemach, zawsze służą do znalezienia optimum pewnego podproblemu. Stąd także dość duża liczba zastosowań

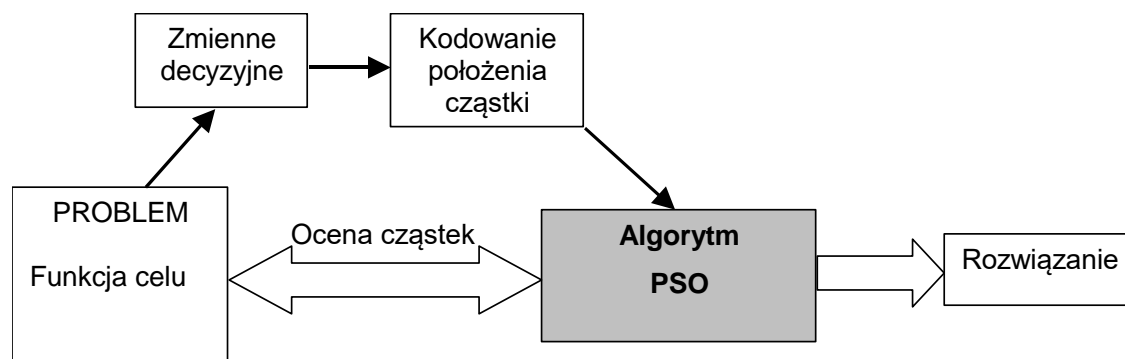
hybrydowych, w których algorytm PSO optymalizuje parametry działania innej metody inteligencji obliczeniowej.

Prognozowanie

Sposób wykorzystania algorytmów PSO w prognozowaniu jest taki sam, jak algorytmów ewolucyjnych (rys 3.9), co oznacza, że algorytm PSO służy do optymalizacji pewnego modelu prognostycznego. Najczęściej optymalizacji podlegają wartości wag połączeń sieci neuronowej. Omawiane algorytmy mogą też zostać wykorzystane do optymalizacji metod statystycznych.

Optymalizacja

Tak jak w przypadku algorytmów ewolucyjnych, optymalizacja to najszersza grupa zastosowań algorytmów PSO. Implementację algorytmu PSO do rozwiązania określonego problemu można zrealizować podobnie, jak ma to miejsce w przypadku algorytmów ewolucyjnych (rys. 3.11). Jedyną różnicą jest to, że zamiast pojęcia osobnika i jego kodowania, wprowadza się pojęcie cząstki i jej położenia w przestrzeni (rys. 3.16).



Rysunek 3.16 Ogólna idea wykorzystania algorytmu PSO w optymalizacji

Szczególnie w przypadku problemów o ciągłych zmiennych decyzyjnych, sposób przejścia pomiędzy zmienną decyzyjną a współrzędną cząstki jest prosty i naturalny. Natomiast w odniesieniu do innych problemów, zwłaszcza kombinatorycznych, należy szukać bardziej wyrafinowanych metod kodowania. Z tego też powodu najwięcej aplikacji algorytmów PSO w elektroenergetyce dotyczy optymalizacji zadań o zmiennych ciągłych. Zostały one wykorzystane między innymi w następujących problemach:

1. Optymalizacja konstrukcji przekładników prądowych

2. Optymalizacja konstrukcji maszyn elektrycznych
3. Optymalizacja strategii gry na rynku energii
4. Optymalizacja lokalizacji łączników w sieci rozdzielczej
5. Optymalizacja rozptyłu mocy biernej
6. Optymalizacja rozptyłu mocy czynnej
7. Optymalizacja lokalizacji/wielkości urządzeń STATCOM i FACTS w sieci
8. Optymalizacja konfiguracji sieci rozdzielczej
9. Optymalizacja harmonogramu prac eksploatacyjnych
10. Optymalizacja lokalizacji/wielkości generacji rozproszonej w sieci
11. Optymalizacja rozwoju sieci
12. Optymalizacja ekonomicznego rozdziału obciążeń
13. Optymalizacja harmonogramu wytwarzania w systemie ze źródłami wiatrowymi i szczytowo-pompowymi

Identyfikacja

Algorytmy PSO mają bardzo duży, podobny do algorytmów ewolucyjnych, potencjał aplikacyjny w ogólnie pojętej identyfikacji. Analogicznie do AE są także stosowane do optymalizacji modeli lub optymalizacji, bądź wspomagania, działania innej metody obliczeniowej. Obecnie liczba zastosowań PSO na tym polu jest niewielka – prawdopodobnie ze względu na relatywną nowość metody. Mimo tego zastosowania są dość ciekawe. Za przykład mogą tu służyć: estymacja stanu systemu, estymacja obciążeń, redukcja rzędu modelu podsystemu, dynamiczna ekwiwalentacja, diagnostyka transformatorów.

Sterowanie

Mimo tego, że algorytmy PSO charakteryzują się zwykle większą szybkością działania niż algorytmy ewolucyjne, to nie są najlepszym rozwiązaniem do stosowania w problemach o sztywnych ograniczeniach czasowych. Natomiast mogą zostać wykorzystane do optymalizacji parametrów ogólnie pojętego algorytmu sterowania tak, by jak najlepiej realizował on postawione przed nim zadania. Takie podejście prezentowane jest w następujących aplikacjach:

1. Stabilizacja systemu elektroenergetycznego
2. Automatyka zabezpieczeniowa linii
3. Automatyka przeciwwawaryjna i restytucyjna

4. Regulacja źródeł wytwórczych

5. Regulacja silników elektrycznych

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 $f_* = 0$ osiągnięte 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 $f_* = 0$ osiągnięte 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 $f_* = 0$ osiągnięte 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 $f_* = 0$ osiągnięte 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 $f_* = 0$ osiągnięte 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 $f_* \approx -418.9829 \cdot n$ osiągane jest dla $x_* = (420.9687, 420.9687, \dots, 420.9687)$.

W celu zapewnienia dodatniości funkcji dla wszystkich numerów wymiarów oraz zmiennych, następująca modyfikacji 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_PSO.bxt, umożliwiającym zbadanie wpływu każdej zmiennej na efekty symulacji. Kombinacje pozostają takie same dla każdej funkcji testowej.
- Porównanie wyników na podstawie plików report_PSO_auto.txt, w tym ocena jakości rozwiązania znalezionego oraz analiza zbieżności