

Particle Swarm Optimization

Szymon Iwanek, Krzysztof Krzysztofik
[siwanek, kkrzysztofik]@wi.zut.edu.pl

WYDZIAŁ INFORMATYKI
ZACHODNIOPOMORSKI UNIwersYTET TECHNOLOGICZNY W SZCZECINIE
UL. ŻOŁNIERSKA 49, 71-210 SZCZECIN









Opis tematu

- W 1995 James Kennedy i Russell Eberhart opublikowali prace o tytule Particle Swarm Optimization
- Zaprezentowali oni nową metodę optymalizacji ciągłych nieliniowych funkcji.
- PSO jest to technika ewolucyjna wzorowana na sposobie poruszania się i zachowania ławic ryb, stad zwierząt, rojów owadów

Opis algorytmu

W każdej iteracji algorytmu wektor prędkości oraz nowa pozycja cząsteczki wyliczane są w następujący sposób:

$$v_{n+1} = w * v_n + c_1 * r_1 * (p_{best} - x_n) + c_2 * r_2 * (g_{best} - x_n) \quad (1)$$

$$x_{n+1} = x_n + v_{n+1} \quad (2)$$

gdzie:

- v_n - wektor prędkości w iteracji n
- x_n - pozycja cząstki w iteracji n
- p_{best} - najlepsza znaleziona pozycja dla danej cząstki
- g_{best} - najlepsza znaleziona pozycja w całym roju
- c_1 - waga poznawcza lub personalna
- c_2 - waga społeczna lub globalna
- w - współczynnik inercji
- r_1, r_2 - losowa liczba z przedziału $[0,1)$

Startowe pozycje cząstek wyliczane są w następujący sposób:

$$x_0 = x_{min} + r_1(x_{max} - x_{min}) \quad (3)$$

gdzie:

- x_{min} - dolne ograniczenie przestrzeni rozwiązań
- x_{max} - górne ograniczenie przestrzeni rozwiązań
- r_1 - losowa liczba z przedziału $(0, 1)$

Wektory prędkości cząstek inicjowane są losowo liczbami o niewielkiej wartości. Sterowanie algorytmem będzie polegało na zmianie współczynników c_1, c_2 oraz w .

Sugerowane wartości parametrów

Wzorując się na badaniach jednego z twórców metody:

- i - ilość iteracji - w typowych zastosowaniach wartości z przedziału $\langle 1000, 100000 \rangle$
- S - liczebność roju - w eksperymentach zastosowano wartości z przedziału $\langle 10, 50 \rangle$
- c_1 - waga poznawcza lub personalna - 1.49445
- c_2 - waga społeczna lub globalna - 1.49445
- w - współczynnik inercji - 0.729, może też być zmienny, malec wraz ze wzrostem licznika iteracji

Pseudokod algorytmu

```
1 for( kazda czasteczka w roju )
2   Zainicjuj losowo czasteczke (polozenie i predkosc)
3
4 do {
5   for( kazda czasteczka w roju ) {
6     Policz wartosc dopasowania
7     if (wartosc dopasowania > pbest)
8       pbest = wartosc dopasowania
9   }
10  [...]
11 } while (nie jest spelniony warunek stopu)
```

Pseudokod algorytmu

```
1 do  
2 {  
3     [...]   
4 gbest = czastka z najlepszym dopasowaniem pbest w calym  
     roju  
5 for( kazda czasteczka w roju )  
6 {  
7 Policz nowa szybkoosc czasteczki  
8 Zmodyfikuj pozycje czasteczki  
9 }  
10 } while (nie jest spelniony warunek stopu)
```

Podsumowanie

Dziękujemy za uwagę