

1. Algorytm optymalizacji rojem cząstek

Algorytm ten bazuje na zachowaniu całej populacji, w której istnieje możliwość komunikowania się między osobnikami-cząstkami i dzielenia się informacjami, przy czym każda cząstka posiada określone położenie i prędkość.

Parametry i -tej cząstki, takie jak położenie oraz prędkość, można przedstawić w postaci wektorów $x_i = [x_{i1}, x_{i2}, \dots, x_{id}]$ oraz $v_i = [v_{i1}, v_{i2}, \dots, v_{id}]$, gdzie d oznacza wymiar przestrzeni. Najlepsza pozycja cząstki $p_i = [p_{i1}, p_{i2}, \dots, p_{id}]$ odpowiada najlepszej uzyskanej dotychczas wartości funkcji celu, zaś najlepsza pozycja cząstki-przywódcy w całym roju określona jest jako $p_d = [p_{d1}, p_{d2}, \dots, p_{dd}]$.

Algorytm optymalizacji rojem cząstek można przedstawić w kilku etapach:

Etap 1: Losowa inicjalizacja pozycji i prędkości początkowych cząstek.

Etap 2: Ocena położenia cząstek za pomocą funkcji dopasowania.

Etap 3: Porównanie zachowania każdej cząstki z jej najlepszym dotychczasowym zachowaniem.

Etap 4: Uaktualnienie prędkości każdej cząstki w każdym kroku k :

$$v_i(k) = \omega v_i(k-1) + c_1 r_1 [p_i(k-1) - x_i(k-1)] + c_2 r_2 [p_d(k-1) - x_i(k-1)] \quad (1)$$

gdzie ω oznacza współczynnik inercji ruchu cząstki, c_1 to parametr określający zaufanie do kierunku swojego najlepszego położenia, c_2 to wskaźnik zaufania do położenia swoich sąsiadów, r_1 oraz r_2 to losowe liczby o rozkładzie równomiernym w przedziale $[0, 1]$.

Etap 5: Uaktualnienie położenia każdej cząstki:

$$x_i(k) = x_i(k-1) + v_i(k) \quad (2)$$

2. Algorytmy pszczele

Ogólną strukturę algorytmu pszczelego można przedstawić następująco:

Etap 1: Losowa inicjalizacja n rozwiązań początkowych (pszczół zwiadowców).

Etap 2: Obliczenie funkcji celu dla całej populacji.

Etap 3: Dopóki niespełnione jest kryterium stopu (zadana liczba iteracji) należy:

- wybrać m sąsiedztw do przeszukiwania,
- zrekrutować pszczoły do wybranych miejsc (liczba pszczół nep do najlepszych e miejsc),
- wyliczyć funkcję celu,
- wybrać najlepszą pszczołę w danym miejscu (najlepsze lokalne rozwiązanie),
- przypisać pozostałe pszczoły do losowych poszukiwań i wyliczyć ich funkcje dopasowania.

Etap 4: Jeśli spełnione kryterium stopu – wyznaczyć najlepsze rozwiązania.

3. Algorytm świetlika

W 2007 roku Xin-She Yang z Uniwersytetu Cambridge, opracował algorytm w oparciu o zachowanie świetlików. I

Każdy świetlik ma swoją atrakcyjność β opisaną poprzez funkcję odległości między dwoma dowolnymi świetlikami:

$$\beta(r) = \beta_0 e^{-\gamma r^m}, \quad m \geq 1, \quad (3)$$

gdzie β_0 oznacza atrakcyjność w $r = 0$ zaś γ jest współczynnikiem absorpcji światła.

Odległość między dwoma świetlikami i oraz j w pozycjach x_i i x_j jest określona jako:

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2}, \quad (4)$$

gdzie d oznacza liczbę wymiarów.

Ruch świetlika i jest określony przez następującą formułę:

$$x_i = x_i + \beta_0 e^{-r_{ij}^2} (x_j - x_i) + \alpha \left(\text{rand} - \frac{1}{2} \right), \quad (5)$$

gdzie pierwszy składnik określa bieżącą pozycję świetlika i , drugi składnik określa atrakcyjność, trzeci składnik używany jest w przypadku losowego przemieszczania (rand jest liczbą losową z zakresu $[0, 1]$, zaś $\alpha \in (0, 1)$). W większości przypadków $\beta_0 = 1$ i $\gamma = 1$.

Ogólna struktura algorytmu jest następująca :

Etap 1: Inicjalizacja parametrów algorytmu (n , β_0 , γ , liczba iteracji, α).

Zdefiniowanie funkcji celu $f(x)$, $x = (x_1, \dots, x_d)^T$.

Wygenerowanie początkowej populacji świetlików.

Intensywność światła i -tego świetlika I_i jest określona przez funkcję celu $f(x_i)$.

Etap 2: Dopóki nie jest spełniony warunek stopu (zadana liczba iteracji):

dla wszystkich n świetlików należy:

jeśli ($I_j > I_i$) to wykonać ruch świetlika i w kierunku świetlika j ,

wyznaczyć atrakcyjność,

znaleźć nowe rozwiązanie i uaktualnić intensywność światła.

Ocena świetlików i znalezienie najlepszego.

Etap 3: Spełniony warunek stopu - wskazanie świetlika z najwyższą intensywnością światła, wizualizacja wyników.

Literatura

1. Eberhart R., Shi Y., Kennedy J.: *Swarm Intelligence*. Morgan Kaufman, San Francisco, 2001.
2. Filipowicz B., Chmiel W., Kadłuczka P.: *Ukierunkowane przeszukiwanie przestrzeni rozwiązań w algorytmach rojowych*. Automatyka, półrocznik AGH, 13(2), 2009.
3. Filipowicz B., Kwiecień J.: *Algorytmy stadne w optymalizacji problemów przydziału przy kwadratowym wskaźniku jakości (QAP)*. Automatyka, półrocznik AGH, 15(2), 2011.
4. Karaboga D., Akay B.: *Artificial Bee Colony (ABC), Harmony Search and Bees Algorithms on numerical optimization*. Artykuł dostępny na stronie (dostęp on-line 28.08.2011): <http://conference.iproms.org/sites/conference.iproms.org/files/IPROMSABCv2.pdf>
5. Kennedy J., Eberhart R.: *Particle Swarm Optimization*. Materiały IEEE International Conference on Neural Networks, 4, 1942-1948, 1995.
6. Łukasik S, Żak S.: *Firefly algorithm for continuous constrained optimization task*. Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems LNCS, 5796, 97-106, 2009.
7. Pham D.T., Ghanbarzadeh A., Koc E., Otri S., Rahim S., Zaidi M.: *The Bees Algorithm – A Novel Tool for Complex Optimisation Problems*. Technical Note, Manufacturing Engineering Centre, Cardiff University, UK, 2005.
8. Trojanowski K.: *Metaheurystyki praktycznie*. Wydawnictwo WIT, Warszawa 2005.
9. Yang X.S.: *Nature-Inspired Metaheuristic Algorithms*. Luniver Press, 2008.
10. <http://www.mathworks.com/matlabcentral/fileexchange/29693-firefly-algorithm>