1 Opis algorytmu

„Algorytm stada cząsteczek naśladuje ludzkie (albo owadzie) zachowania stadne. Podczas zdobywania własnego doświadczenia cząsteczki wzajemnie na siebie oddziałują, a członkowie populacji stopniowo przenoszą się w coraz lepsze obszary przestrzeni rozwiązań.”

J. Kennedy i R. Eberhart

PSO (Particle Swarm Optimization) to jedna z szerokiej gamy metod inteligencji stadnej stosowana w rozwiązywaniu problemów optymalizacji globalnej. Żeby lepiej zrozumieć PSO należy przybliżyć pojęcie **inteligencji stadnej**. Jest to technika sztucznej inteligencji, bazująca na analizie i studiach zachowań kolektywnych, które sa zdecentralizowanymi, samoorganizującymi się systemami.

Oparte na inteligencji stadnej techniki mają liczne zastosowania typu: kontrolowanie bezzałogowych pojazdów, kontrola nanobotów eliminujących komórki rakowe czy tworzenie odwzorowań planetarnych. PSO samo w sobie wykorzystane jest głównie przy rozwiązywaniu problemów typu:

- kompresja cylindryczna,
- serwomechanizmy,
- optymalizacja numeryczna,
- estymacja baterii,
- trenowanie sieci neuronowych,
- kompresja powietrza.

Algorytm ten stał się w ostatnich czasach bardzo popularny przede wszystkim dlatego, że na jego działanie wpływa mała liczba parametrów.

Każda cząstka $a \in 1, 2, ..., A$ jest określana za pomocą jej pozycji X_{ai} i prędkości $V_a i$ dla każdego z wymiarów $i \in 1, 2, ..., n$. W każdej iteracji wyznaczamy nowe prędkości i pozycje cząstek korzystając ze wzorów:

$$V_{ai}(t+1) = \omega V_{ai}(t) + C_1\phi_1(P_{lai} - X_{ai}(t)) + C_2\phi_2(P_{gi} - X_{ai}(t)) \tag{1}$$

$$X_{ai}(t+1) = X_{ai}(t) + V_{ai}(t+1) \tag{2}$$

gdzie:

- $0 < \phi_1, \phi_2 < \phi_{max}$ - losowe wartości od 0 do 1
- C_1, C_2 - stałe przyspieszenia losowane z przedziału od 0 do 1
- ω - współczynnik bezwładności
- P_{la} - najlepsze rozwiązanie znalezione przez cząstkę
- P_g - najlepsze globalne rozwiązanie

fig1.PNG

Rysunek 1: Podstawowe pojęcia PSO w 2D

Warto zaznaczyć również, że PSO to metoda metaheurystyczna, więc możemy optymalizować nią nieróżniczkowalne, a nawet nieciągłe funkcje. Dzięki temu nadaje się do problemów dynamicznych, z dynamicznie zmieniającą się dziedziną.

Biblioteka nazwana przeze mnie `OptiPSO` będzie służyła do rozwiązania dowolnego problemu, który można sformułować jako minimalizację ciągłej funkcji rzeczywistej $y = f(x)$, gdzie x jest wektorem liczb rzeczywistych, z ograniczeniami kostkowymi. Rozwiązanie będzie obejmować również zestaw testów dla typowych funkcji stosowanych do testowania programów optymalizujących (Rosenbrocka, Rastrigina, itp).