

# Przeszukiwanie i optymalizacja - dokumentacja wstępna

Maksym Bieńkowski, Jędrzej Grabski

01.12.2024

**Temat projektu:** Algorytm roju cząstek z modyfikacjami dotyczącymi współczynnika bezwładności

## 1 Analiza problemu

Ideą Algorytmu Roju Cząstek (PSO - Particle Swarm Optimalization), jest symulowanie populacji ("roju"), która rozwija się na podstawie wiedzy pojedynczych osobników ("cząstek") oraz pewnej wiedzy dzielonej. Każda z cząstek posiada swoją prędkość oraz pozycję w przestrzeni rozwiązań. Ponadto zapamiętywane jest najlepsze rozwiązanie znalezione do tej pory przez każdą z cząstek (optimum lokalne, niewspółdzielone z resztą populacji), a także najlepsze rozwiązanie z całego roju (optimum globalne, współdzielone przez wszystkie cząstki).

Prędkość  $i$ -tej cząstki w epoce  $k + 1$  dana jest następującym wzorem:

$$V_i(k + 1) = wV_i(k) + \phi_p r_1 (P_i(k) - X_i(k)) + \phi_g r_2 (P_g(k) - X_i(k))$$

gdzie  $w$  oznacza współczynnik bezwładności,  $P_i$  położenie optimum lokalnego,  $P_g$  położenie optimum globalnego, współczynniki  $r_1$  i  $r_2$  losowane są z rozkładem  $U[0, 1]$ , a  $\phi_p$  oraz  $\phi_g$  oznaczają kolejno parametryzowane współczynniki wagi - poznawczy i społeczny. Na podstawie powyższego wzoru obserwujemy, że wektor prędkości tworzony jest na podstawie trzech składowych, a współczynnik bezwładności określa wagę składowej będącej prędkością w poprzedniej iteracji. Im mniejsza wartość tego współczynnika, tym bardziej zwrotne i skłonne do eksploatacji są cząstki. Ze zwiększeniem wartości współczynnika bezwładności wiąże się natomiast większa skłonność do eksploracji przestrzeni, co może jednak skutkować "przeprzeliwaniem" optimów lokalnych.

## 2 Propozycja rozwiązania

Problem sformułowany w poprzedniej sekcji spróbujemy zniwelować poprzez wprowadzenie dynamicznej zmiany współczynnika bezwładności, uzależniając go od liczby wykonanych iteracji. Współczynnik będzie stopniowo zmniejszany się w miarę pracy algorytmu. Umożliwi to skupienie się na eksploracji w początkowej fazie algorytmu, a następnie bardziej precyzyjne zbieganie wokół optimów pod koniec pracy.

## 3 Przyjęte założenia

### Przestrzeń rozwiązań

Zakładamy, że przestrzeń rozwiązań jest ciągła, ograniczona i wielowymiarowa, a wartości funkcji celu są dobrze zdefiniowane w całej przestrzeni.

### Funkcja celu

Funkcja celu jest różnorodna, tj. może być jedno- lub wielomodalna, aby przetestować algorytm w różnych warunkach.

### Początkowa populacja

Pozycje cząstek w roju są inicjalizowane losowo z rozkładem jednostajnym w granicach przestrzeni rozwiązań. Początkowe prędkości cząstek będą losowane z rozkładem jednostajnym na bazie ograniczeń przestrzeni przeszukiwań.

### Parametry algorytmu

- Liczba cząstek w roju oraz liczba iteracji są ustalane na początku i pozostają stałe w trakcie pracy algorytmu.
- Parametry algorytmu dobrane zostaną na podstawie literatury.
- Dynamiczny współczynnik bezwładności: Zmiana współczynnika bezwładności następuje zgodnie z wzorem  $w' = w * u^{-k}$ , gdzie  $w \in [0, 1]$  to bazowy współczynnik bezwładności,  $u \in [1.0001, 1.005]$  to siła wytracania wartości współczynnika, a  $k$  to numer iteracji.

## 4 Sposób badania jakości rozwiązania

### Porównanie algorytmów

Algorytm PSO z dynamicznym współczynnikiem bezwładności zostanie porównany z klasyczną wersją tego algorytmu (ze stałym współczynnikiem). Testy zostaną przeprowadzone na dobrze znanych benchmarkach funkcji optymalizacyjnych w zróżnicowanych ilościach wymiarów.

### Kryteria oceny

- Jakość rozwiązania: Różnica wartości funkcji celu algorytmu od optimum globalnego dla danego problemu optymalizacji po określonej liczbie iteracji.
- Szybkość zbieżności: Analiza tempa zbliżania się do optimum, mierzona przez ilość iteracji po której zostało osiągnięte otoczenie  $\epsilon$  optimum globalnego.

## Proces badawczy

Testy zostaną przeprowadzone na przestrzeniach o różnych wymiarach, aby ocenić wpływ wymiarowości na działanie algorytmu oraz efektywność wprowadzonej modyfikacji.

## Wizualizacja

Dla każdego przeprowadzonego badania przedstawiony będzie wykres wartości funkcji celu w  $P_g$  na przestrzeni iteracji. Ponadto zamierzamy zwizualizować trajektorię populacji w niskowymiarowej przestrzeni na wykresie poziomym.

## 5 Środowisko

- Projekt będzie realizowany w środowisku Python.
- Narzędziem do zarządzania zależnościami oraz środowiskiem wirtualnym będzie *PDM*.
- Do wizualizacji przebiegów działań algorytmów zostanie użyta biblioteka *matplotlib* lub *seaborn*

## 6 Bibliografia

- Particle swarm optimization. (1995). IEEE Conference Publication | IEEE Xplore. <https://ieeexplore.ieee.org/document/488968>
- Jiao, B., Lian, Z., & Gu, X. (2006). A dynamic inertia weight particle swarm optimization algorithm. *Chaos Solitons & Fractals*, 37(3), 698–705. <https://doi.org/10.1016/j.chaos.2006.09.063>