

**RandomNumbers**

to funkcja zwracająca listę n losowych liczb z podanego przedziału (n to wymiar wektora)

START

**Particle**

to klasa reprezentująca cząsteczkę o polach typu lista: pozycja, najlepsza pozycja oraz prędkość

funkcja kryterialna, przedział liczbowy, wymiar wektora,  
generationCount (liczba generacji),  
populationCount (liczba osobników)

SPR – współczynnik korekcji najlepszej pozycji roju  
PPR – współczynnik korekcji najlepszej pozycji cząsteczki  
VR – współczynnik korekcji prędkości

Stworzenie populacji cząsteczek o losowych pozycjach i zerowej prędkości:  
population = [Particle(RandomNumbers) populationCount razy]

Posortuj cząsteczki pozycjami rosnąco/malejąco wg f. kryterialnej  
najPozRoju = population[0].pozycja (najlepsza dotychczas pozycja roju)  
t = 0

t += 1

Czy  $t \leq \text{generationCount}$ ?

NIE

TAK

**RUCH CZĄSTECZEK**

Dla każdej cząsteczki w *population*:

Dla każdej j-tej współrzędnej pozycji cząsteczki:

$$\alpha = \text{random}(0;1), \beta = \text{random}(0;1)$$
$$\text{prędkość}[j] = \text{VR} * \text{prędkość}[j]$$
$$+ \text{PPR} * \alpha * (\text{najPozycja}[j] - \text{pozycja}[j])$$
$$+ \text{SPR} * \beta * (\text{najPozRoju}[j] - \text{pozycja}[j])$$
$$\text{pozycja}[j] += \text{prędkość}[j]$$

Jeśli wartość **funkcja(pozycja)** jest lepsza\*  
od wartości **funkcja(najPozycja)**,  
to przypisz *najPozycja* w cząsteczce wartość *pozycja*.

\*tj. mniejsza w przypadku szukania min i większa w przyp. max

**WYBÓR LIDERA ROJU**

Posortuj cząsteczki pozycjami rosnąco/malejąco wg f. kryterialnej.

Jeśli wartość **funkcja(population[0].pozycja)** jest lepsza\*

od wartości **funkcja(najPozRoju)**,

to przypisz zmiennej *najPozRoju* wartość *population[0].pozycja*.

\*tj. mniejsza w przypadku szukania min i większa w przyp. max

Zwróć *najPozRoju*  
i **funkcja(najPozRoju)**

STOP