

RandomNumbers

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

START

Firefly

to klasa reprezentująca świetliki o polu *pozycja* (typu lista)

funkcja kryterialna,
przedział liczbowy, wymiar wektora,
generationCount (liczba generacji),
populationCount (liczba osobników)
 γ - współczynnik absorpcji światła
 β_0 - atrakcyjność dla zerowego dystansu
 α - współczynnik randomizacji

Stwórz populację świetlików o losowych współrzędnych:
population = [Firefly(RandomNumbers) populationCount razy]
t = 0

t += 1

Czy t ≤ generationCount?

NIE

TAK

PRZCIĄGANIE SIĘ ŚWIETLIKÓW

Dla każdego świetlika *fireflyOne* w *population* wykonaj wszystkie poniższe czynności:

Dla każdego świetlika *fireflyTwo* w *population* wykonaj wszystkie poniższe czynności:

Oblicz *dystans* pomiędzy pozycjami świetlików 1 i 2 zgodnie z metryką Kartezjusza

Oblicz atrakcyjność lokalizacji: $\text{attr} = \beta_0 \cdot e^{-\gamma \cdot \text{dystans} \cdot \text{dystans}}$

pos1 = pozycja *fireflyOne*, pos2 = pozycja *fireflyTwo*, nowaPozycja = pos1

Przesuń *fireflyOne* na każdej k-tej współrzędnej w stronę *fireflyTwo*:

$\text{nowaPozycja}[k] += \text{attr} \cdot (\text{pos2}[k] - \text{pos1}[k]) + N(0,1) \cdot \alpha$

Jeśli wartość **funkcja**(*nowaPozycja*) jest lepsza*
od wartości **funkcja**(*fireflyOne.pozycja*),
to przypisz polu *pozycja* w świetliku wartość *nowaPozycja*.

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

WYBÓR NAJATRAKCYJNIEJSZEGO ŚWIETLIKA

Posortuj *population* pozycjami rosnąco/malejąco wg f. kryterialnej
bestPosition = population[0].position[:]

Zwróć *bestPosition*
i **funkcja**(*bestPosition*)

STOP