

# 1 Algorytm kukułki

Start

Wczytaj parametry  $population\_size$ ,  $iteration\_count$ ,  $\beta$ ,  $\gamma$ ,  $\delta$ ,  $p_a$

Wczytaj funkcję celu do minimalizacji  $cost\_function$  oraz badany obszar  $domain$

$population \leftarrow$  pusta lista

**Dla**  $i \leftarrow 0$  **do**  $population\_size$  **wykonuj**

$position \leftarrow$  losowy punkt w badanym obszarze

    Dodaj  $position$  do listy  $population$

**koniec**

$best\_nest \leftarrow$  pozycja z  $population$  o najmniejszej wartości funkcji celu

$lowest\_cost \leftarrow cost\_function(best\_nest)$

**Dla**  $i \leftarrow 0$  **do**  $iteration\_count$  **wykonuj**

**Dla**  $j \leftarrow 0$  **do**  $current\_population$  **wykonuj**

$current\_cuckoo \leftarrow population[j]$

$nest\_position \leftarrow$  nowa pozycja utworzona na bazie  $current\_cuckoo$  na podstawie wzoru (1)

        Ogranicz współrzędne  $nest\_position$  do badanego obszaru

**Jeżeli**  $cost\_function(nest\_position) < cost\_function(current\_cuckoo)$  **to**

$population[i] \leftarrow nest\_position$

**koniec**

**koniec**

    Posortuj  $population$  według rosnącej wartości funkcji celu

**Jeżeli**  $cost\_function(population[0]) < lowest\_cost$  **to**

$best\_nest \leftarrow population[0]$

$lowest\_cost \leftarrow cost\_function(best\_nest)$

**koniec**

$nests\_to\_abandon \leftarrow \lfloor p_a * population\_size \rfloor$

**Dla**  $j \leftarrow population\_size - nests\_to\_abandon$  **do**  $population\_size$  **wykonuj**

$position \leftarrow$  losowy punkt w badanym obszarze

$population[i] \leftarrow position$

**koniec**

**koniec**

Wypisz  $best\_position$  oraz  $lowest\_cost$

Stop

## 2 Wzory

### 2.1 Pozycja gniazda

$$n_j = \Phi_j * L(\beta, \gamma, \delta)$$
$$L(\beta, \gamma, \delta) = \sqrt{\frac{\gamma}{2 * \pi}} * \frac{\exp\left(\frac{-\gamma}{2 * (\beta - \delta)}\right)}{(\beta - \delta)^{\frac{3}{2}}} \quad (1)$$

Gdzie  $\Phi_j$  jest losową liczbą z przedziału  $[0, 1]$  dla każdego  $j$ .