

1 Algorytm kukułki

Start
Wczytaj parametry $population_size$, $iteration_count$, β , γ , δ , 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**
| | | $population[j] \leftarrow nest_position$
| | **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
| | $population[j] \leftarrow position$
koniec
koniec
Wypisz $best_nest$ 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 Φ_j jest losową liczbą z przedziału $[0, 1]$ dla każdego j .