

Optymalizacja rojem cząstek funkcji testowych

Joanna Kołodziejczyk

1 Problem do rozwiązania

Celem laboratorium jest implementacja algorytmu PSO.

Optymalizacja ma zostać wykonana dla testowego problemu optymalizacji globalnej w przestrzeni liczb rzeczywistych. Poniżej znajduje się lista funkcji, na których należy przeprowadzić testy:

1. RASTRIGIN FUNCTION

- <http://www.sfu.ca/~ssurjano/rastr.html>,
- $domain = [-5.12, 5.12]$
- Globalne minimum $f(0, 0, \dots, 0) = 0$
- Wymiarowość: $\{2, \dots, d\}$

2. GRIEWANK FUNCTION

- <http://www.sfu.ca/~ssurjano/griewank.html>,
- $domain = [-600, 600]$
- Globalne minimum $f(0, 0, \dots, 0) = 0$
- Wymiarowość: $\{2, \dots, d\}$

3. SPHERE FUNCTION

- <http://www.sfu.ca/~ssurjano/spheref.html>,
- $domain = [-5.12, 5.12]$
- Globalne minimum $f(0, 0, \dots, 0) = 0$
- Wymiarowość $\{2, \dots, d\}$

4. ZAKHAROV FUNCTION

- <http://www.sfu.ca/~ssurjano/zakharov.html>,
- $domain = [-5, 10]$
- Globalne minimum $f(0, 0, \dots, 0) = 0$
- Wymiarowość: $\{2, \dots, d\}$

5. EASOM FUNCTION

- <http://www.sfu.ca/~ssurjano/easom.html>,
- $domain = [-100, 100]$
- Globalne minimum $f(\pi, \pi) = -1$
- Wymiarowość: 2

6. STYBLINSKI-TANG FUNCTION

- <http://www.sfu.ca/~ssurjano/stybtang.html>,
- $domain = [-5, 5]$
- Globalne minimum $f(-2.903534, \dots, -2.903534) = -39.16599d$, gdzie d jest wymiarowością problemu.
- Wymiarowość: $\{2, \dots, d\}$

2 Algorytm

2.1 Pseudokod algorytmu PSO

```
1: swarm_initialize(swarm_size)
2:  $t \leftarrow 0$ 
3: while terminal_condition not TRUE do
4:   for  $i = 1$  to swarm_size do
5:     if  $f(x_i) < f(\textit{personal\_best}_i)$  then
6:        $\textit{personal\_best}_i \leftarrow x_i$ 
7:     end if
8:     if  $f(\textit{personal\_best}_i) < f(\textit{global\_best})$  then
9:        $\textit{global\_best} \leftarrow \textit{personal\_best}_i$ 
10:    end if
11:  end for
12:  for  $i = 1$  to swarm_size do
13:    for  $j = 1$  to  $d$  do
14:       $v_{i,j} \leftarrow \omega \cdot v_{i,j} + \phi_p \cdot rU(0, 1) \cdot (\textit{personal\_best}_{i,j} - x_{i,j}) + \phi_g \cdot rU(0, 1) \cdot (\textit{global\_best}_j - x_{i,j})$ 
15:       $x_{i,j} \leftarrow x_{i,j} + v_{i,j}$ 
16:    end for
17:  end for
18:   $t \leftarrow t + 1$ 
19: end while
20: return swarm
```

2.2 Istotne elementy algorytmu

1. Kodowanie cząstki:

- x — wektor cech cząstki (współrzędne punktu w przestrzeni \mathbb{R}^n) - zakres z dziedziny zadanej funkcji (*domain*). Liczba elementów w wektorze to wymiarowość funkcji testowej d .
- v — wektor prędkości dla każdej cząsteczki. Liczba elementów w wektorze to wymiarowość funkcji testowej d .
- *personal_best* — najlepsze położenie cząstki x w całym procesie poszukiwania. Liczba elementów w wektorze to wymiarowość funkcji testowej d .
- f — poziom przystosowania cząstki - zadana funkcja testowa do optymalizacji. f jest skalar, wyliczoną wartością ze wzoru na funkcję testową.

2. Parametry początkowe algorytmu

Przed uruchomieniem algorytmu należy arbitralnie podać wielkości parametrów:

- *swarm_size* — liczba cząstek w roju: zalecane wartości od 10 do 50
- ϕ_p — parametr odpowiedzialny za indywidualność cząstki. Jeżeli wartość duża, to cząstki chętnie podążają za swoimi dotychczas najlepszymi miejscami $0 < \phi_p < 4$.
- ϕ_g — parametr odpowiedzialny za stadność cząstki. Jeżeli jest duża wartość to silne przyciąganie cząstek przez najlepszego osobnika w roju $0 < \phi_g < 4$.
- ω — inercja - wpływa na wartość prędkości. $\omega = 1$ — cząsteczki nie zwalniają, $\omega < 1$ cząsteczki zwalniają z upływem czasu, $\omega > 1$ cząsteczki przyspieszają.
- *v_max* — maksymalna prędkość cząstki (najczęściej szerokość dziedziny funkcji optymalizowanej) np. *domain* = $\{-5 \dots 5\}$ to *v_max* = 10.

3. Warunek zakończenia *teminal_condition* powinien sprawdzać zadaną maksymalną liczbę kroków algorytmu i osiągnięcie znanego dla funkcji testowej minimum globalnego przy pewnym założonym błędzie np. $f(\text{global_best}) - f_{\text{globalne}} < 0,0001$.

4. Inicjalizacja

Utworzenie roju i wybór pozycji lokalnie najlepszej i globalnie najlepszej

```

for  $i = 1$  to swarm_size do
  for  $j = 1$  to  $d$  do
     $v_{i,j} \leftarrow \text{rand}(-v\_max/3, v\_max/3)$ 
     $x_{i,j} \leftarrow \text{rand}(\text{domain\_min}, \text{domain\_max})$ 
  end for
   $f \leftarrow f(x_i)$ 
  personal_best $_i \leftarrow x_i$ 
  if  $i == 1$  then
    global_best  $\leftarrow x_i$ 
  end if
  if  $f(\text{personal\_best}_i) < f(\text{global\_best})$  then
    global_best  $\leftarrow \text{personal\_best}_i$ 
  end if

```

end for

gdzie:

domain_min i *domain_max* minimalna i maksymalna wartość z dziedziny funkcji optymalizowanej.

5. $rU(0, 1)$ — liczba losowa z rozkładu jednostajnego.

3 Wymagania dla implementacji

Język implementacji algorytmu jest dowolny.

Wymagane elementy programu:

- Do optymalizacji wybrać jedną z podanych na początku funkcji testowych. Wczytać wszystkie parametry z nią związane. Wymiarowość funkcji d jest parametrem programu.
- Zaimplementować algorytm zgodnie z wytycznymi podanymi w punkcie 2.1 i 2.2.
- Program po wykonaniu optymalizacji powinien wyświetlać:
 - położenie znalezione przez algorytm optimum i wartość funkcji w tym punkcie
 - iterację, w której znaleziono rozwiązanie
 - wykres zmienności funkcji celu dla najlepszej cząstki w krokach
 - wykres zmienności średniej funkcji celu w krokach
 - Nieobowiązkowo:
 - * odległość od znanego globalnego optimum (miara Euklidesa)
 - * rozproszenie roju (średni dystans pomiędzy cząstkami w roju) w krokach

Każde własne (rozsądne) propozycje modyfikacji lub wykorzystane z literatury zmiany w algorytmie lub reprezentacji lub prezentacji wyników ponad te, wymagane w instrukcji, będą dodatkowo punktowane.

3.1 Przekazanie programu

Kod z rozwiązaniem proszę wysłać na mail: jkolodziejczyk@zut.edu.pl