

# [PSZT-P] Kiedy się poddać?

Bartosz Świtalski

Piotr Frątczak

24 listopada 2020

## 1 Opis problemu

Algorytmy ewolucyjne w klasycznym wydaniu nie mogą same zdecydować, kiedy zakończyć swoje działanie. W związku z tym należy rozważyć implementację określonego z góry kryterium. Nie istnieje uniwersalne kryterium wykrywania bezcelowości dalszej pracy algorytmu ewolucyjnego. W naszej pracy proponujemy przykładowe rozwiązania, zaimplementujemy je oraz dokonamy analizy ich działania.

## 2 Decyzje projektowe

Optymalizacja zostanie przeprowadzona na funkcjach z [cec2005](#). Zaimplementowana została strategia ewolucyjna ( $\mu + \lambda$ ). Przyjęty budżet możliwych ewaluacji funkcji celu dla pojedynczej próby optymalizacji wynosi  $10000 \cdot \text{wymiarowość zadania}$ . Skupiamy się na wymiarowości  $D = 10$ . Podczas jednego uruchomienia programu dokonujemy uśrednienia wyników z 25 wywołań algorytmu.

## 3 Cele eksperymentu

Implementacja kryteriów przerwań optymalizacji. Dobranie przykładowych parametrów dla odpowiednich kryteriów. Zbadanie wpływu tychże kryteriów na ogólny czas optymalizacji oraz dokładność uzyskanego wyniku (optimum).

## 4 Użycie

```
/when-to-surrender$
```

```
python3 when-to-surrender/main.py  
<funkcja> <kryterium> <p1> <p2> <p3> <p4>
```

### 4.1 Oznaczenia argumentów

<funkcja> - optymalizowana funkcja (dozwolone wartości:  $F4$ ,  $F5$ ,  $F6$ ).

<kryterium> - kryterium przerwania (dozwolone wartości:  $k\text{-iter}$ ,  $sd$ ,  $best\text{-}worst$ ,  $variance$ ).

<p<sub>i</sub>> - kolejne wartości parametru do wcześniej sprecyzowanego kryterium.

### 4.2 Użycie skryptu

```
/when-to-surrender$
```

```
./xscript.sh  
# po wykonaniu  
./xclean.sh
```

### 4.3 Komentarz do użycia

Skrypt umożliwia uruchomienie optymalizacji wszystkich dostępnych funkcji (3) według wszystkich zaimplementowanych kryteriów (4) z wcześniej określonymi parametrami (8 różnych). Pojedyncze uruchomienie skryptu jest dość kosztowne czasowo ( $>2h$ ), więc w celu skrócenia czasu wykonania zalecany jest np. przydział zadań do różnych rdzeni (komenda `taskset`), ale optymalizacja czasu wykonania wielu uruchomień na raz nie jest tematem projektu.

Wygenerowane dane zostaną zapisane do plików w folderze `output`. Do czyszczenia służy skrypt `xclean.sh`.

## 5 Kryteria przerwań

- **$K$ -iterations ( $K$ -iteracji).** Wykorzystane jako kryterium bazowe. Kryterium jest spełnione, jeśli nie ma poprawy wartości funkcji celu przez  $K$  kolejnych iteracji. Należy wybrać odpowiednią wartość  $K$  przy założeniu, że niemożliwe jest uzyskanie lepszego wyniku po  $K$  kolejnych iteracjach.
- **Standard Deviation (Odchylenie Standardowe).** Kryterium jest spełnione, jeżeli odchylenie standardowe wartości każdej z cech osobników obecnej generacji jest mniejsze lub równe niż dane próg  $\epsilon \geq 0$ .

- **Best-worst (Najlepszy-Najgorszy).** Kryterium jest spełnione, gdy różnica funkcji celu między najlepszym i najgorszym osobnikiem jest mniejsza lub równa niż dany próg  $\epsilon \geq 0$ .
- **Fit-Variance (Wariancja Dopasowania).** Kryterium uwzględniające zróżnicowanie funkcji celu wszystkich osobników populacji. Jest spełnione, jeśli wariancja funkcji celu dla wszystkich osobników w pokoleniu jest mniejsza lub równa niż dany próg  $\epsilon$  przy czym  $1 \gg \epsilon \geq 0$ .

## 6 Uzyskane wyniki

Uzyskane wyniki dla uruchomień funkcji według każdego kryterium z różnymi parametrami zostały zebrane w pojedynczy plik `output.pdf` i zostaną omówione w dalszej części tego sprawozdania.

### 6.1 Komentarz do wyników

Przedstawione wyniki dotyczą średniej z 25 uruchomień algorytmu według ustalonego kryterium i ustalonych parametrów.

## 7 Wnioski

### 7.1 Wnioski do $F4$

Najdokładniej jesteśmy w stanie określić optimum globalne dla funkcji  $F4$ . Wykorzystanie budżetu w  $1/3$  gwarantuje dopasowanie rzędu  $10^0$ . Łatwo osiągalne średnie dopasowania rzędu  $10^{-1}$ , co oznacza, że optymalizacja łatwo zmierza w kierunku optimum globalnego.

### 7.2 Wnioski do $F5$

Funkcja  $F5$  jest nieco trudniejsza do optymalizacji. Bardzo rzadko jesteśmy w stanie przeskoczyć siódła w otoczeniu optimum globalnego o rzędzie wielkości  $10^0$ .

### 7.3 Wnioski do $F6$

Najciężej znaleźć optimum globalne funkcji  $F6$ , co prawdopodobnie wiąże się z faktem, że jest to funkcja multimodalna. Nawet dla maksymalnego

wykorzystania budżetu zauważalne są odchylenia standardowe dopasowania rzędu  $10^3$ .

### 7.4 Wnioski do kryterium Wariancji Dopasowania

## 8 Wkład pracy

## 9 Użyte narzędzia


### 9.1 Język programowania

Python 3.8.

#### 9.1.1 Moduły

- [optproblems](#) - zawiera zbiór powszechnie używanych benchmarków. Wykorzystano szczególnie moduł `optproblems.cec2005` zawierający benchmarki CEC 2005, do badania działania zaimplementowanego algorytmu i kryteriów przerwań optymalizacji.
- [random](#) - użycie funkcji do generowania liczb pseudolosowych z rozkładu normalnego i jednostajnego.
- [math](#) - użycie funkcji matematycznych (`exp`, `pierwiastek` itd.).
- [numpy](#) - wykorzystanie struktur do obsługi danych.
- [matplotlib](#) - użyte do graficznej prezentacji danych w postaci wykresów.

## 10 Powiązane linki

 [Repozytorium projektowe](#)

## 11 Bibliografia

GHOEISHI, Seyyede Newsha; CLAUSEN, Anders; JØRGENSEN, Bo Nørregaard. Termination Criteria in Evolutionary Algorithms: A Survey. In: IJCCI. 2017. p. 373-384.

SUGANTHAN, Ponnuthurai N., et al. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. KanGAL report, 2005, 2005005.2005: 2005.