

# Specyfikacja Wstępna

## Zmodyfikowany algorytm ewolucji różnicowej

Monika Żurkowska  
Kacper Sarnacki

15 maja 2016

### 1 Wstęp

Ewolucja różnicowa jest stosunkowo nowym algorytmem optymalizacji numerycznej. Osobnikami są  $L$ -wymiarowe wektory liczb rzeczywistych, w których mutacja polega na perturbacji wybranego wektora o różnicę dwóch innych wektorów pomnożoną przez stały współczynnik.

Algorytm jest bardzo prosty i działa bardzo dobrze na typowych funkcjach testowych. Wymaga tylko trzech parametrów:

$\mu$  - rozmiar populacji  
 $F$  - odpowiedzialny za sterowanie mutacją  
 $c_r$  - sterujący krzyżowaniem

### 2 Pseudokod

```
 $P^0 = P_1^0, P_2^0, P_3^0 \dots P_\mu^0$   
 $t = 0$   
while !stop  
  for  $i \in 1 : \mu$   
     $P_j^t = \text{select}(P^t)$   
     $P_k^t, P_l^t = \text{sample}(P^t)$   
     $M_i^t = P_j^t + F \cdot (P_k^t - P_l^t)$   
     $O_i^t = \text{crossover}(P_i^t, M_i^t)$   
     $P_i^{t+1} = \text{tournament}(P_i^t, O_i^t)$   
   $t = t + 1$ 
```

### 3 Selekcja

Algorytm ewolucji różnicowej operuje na populacji  $\mu$  osobników:  $P_1, P_2, \dots, P_\mu$ , gdzie  $P_i \in \mathbf{R}^L$ . W każdej iteracji  $t$ , dla każdego osobnika  $P_i$  z populacji liczony jest osobnik  $P_j$  będący średnią wszystkich  $\mu$  osobników populacji. Następnie wybierane są losowo osobniki  $P_k, P_l$  takie, że  $i \neq l \neq k$ .

### 4 Mutacja

Wynikiem mutacji jest nowy wektor  $M_i$  (zwany osobnikiem **mutantem**) otrzymany w następujący sposób:

$$M_i = P_j + F \cdot (P_k - P_l) \quad (1)$$

gdzie  $F \in [0, 1]$  jest stałym parametrem zwanym **współczynnikiem amplifikacji** (typowa wartość tego współczynnika wynosi  $F = 0.5$ ). Operacja sumy oraz różnicy polega na dodaniu (odjęciu) odpowiadających sobie składowych wektorów dodawanych (odejmowanych).

### 5 Krzyżowanie

Wynikiem krzyżowania operującego na rodzicu  $P_i$  i mutancie  $M_i$  jest **osobnik próbny**  $O_i$ , który następnie w procesie sukcesji zostanie porównany z osobnikiem  $P_i$ . Każdy element  $O_{i,j} (j = 1, 2, \dots, L)$ , wektora  $O_i$  jest wyznaczany w następujący sposób:

$$O_{i,j} = \begin{cases} M_{i,j} & \text{jeżeli } rnd_j < c_r \text{ lub } j = d \\ P_{i,j} & \text{w przeciwnym przypadku} \end{cases} \quad (2)$$

gdzie  $rnd_j$  jest liczbą losową z przedziału  $[0,1)$  losowaną niezależnie od każdego  $j$ .  $c_r \in [0, 1]$  jest stałym parametrem algorytmu a  $d$  jest losowym numerem elementu wektora losowanego ze zbioru  $1, 2, \dots, L$ .

$c_r$  oznacza prawdopodobieństwo przejścia elementu z wektora mutantu  $M_i$  do wektora próbnego  $O_i$ . Dla  $c_r = 1$  wszystkie elementy wektora próbnego  $O_i$  pochodzą z mutantu  $M_i$  (analogicznie dla  $c_r = 0$  wszystkimi, z wyjątkiem jednego - warunek  $j = d$  - elementami wektora  $M_i$  będą elementy pochodzące od rodzica  $P_i$ ). Operator krzyżowania "miesza" więc losowo elementy rodzica  $P_i$  i mutantu  $M_i$  dając w wyniku wektor próbny  $O_i$ .

Następnie dopasowanie osobnika próbnego  $O_i$  porównywane jest z osobnikiem rodzica  $P_i$  (funkcja *tournament*). Jeśli jest ono lepsze od rodzica (czyli jego wartość funkcji celu jest niższa),  $P_i$  zostaje zastąpione osobnikiem próbnym, w przeciwnym przypadku osobnik  $O_i$  zostaje odrzucony.

## 6 Testowanie

Powyższy algorytm będzie porównywany z klasycznym algorytmem genetycznym. Do porównania użyta zostanie wersja z selekcją losowego osobnika (DE/rand/1/bin) oraz najlepszego w populacji (DE/best/1/bin).

Jako funkcję celu wykorzystamy 28 funkcji benchmarka CEC-2013. Testom poddane zostaną klasyczne algorytmy ewolucji różnicowej oraz zmodyfikowana wersja uwzględniająca wybór średniej populacji w mutacji z kolejno 10, 30 i 50-wymiarowymi wektorami populacji. Każdy test zostanie przeprowadzony 21 razy (dla każdego wymiaru osobno). Następnie dla każdego algorytmu i dla każdej funkcji benchmarka zostanie obliczona tabela zawierająca: najlepsze, najgorsze, średnie oraz medianowe rozwiązanie a także odchylenie standardowe.

Algorytm zatrzyma się po odpowiedniej ilości iteracji -  $10000 * D$ , gdzie  $D$  jest wymiarem wektora (10, 30 i 50), bądź gdy wartość błędu spadnie poniżej  $10^{-8}$ . Przestrzeń przeszukiwania dla każdego wymiaru znajdować się będzie w zakresie  $[-100, 100]^D$ . Pierwsza populacja generowana będzie losowo w przestrzeni przeszukiwania.

Oczekiwanym rezultatem testów jest najlepszy wynik dla algorytmu klasycznego z selekcją najlepszego elementu populacji. Gorszym powinien okazać się algorytm z wyborem elementu średniego, a najgorszy z wyborem elementu losowego.

## Literatura

- [1] <https://elektron.elka.pw.edu.pl/~jarabas/WAE/wyklad10.pdf>
- [2] <https://www.rforge.net/doc/packages/cec2013/cec2013.html>
- [3] [http://www.ntu.edu.sg/home/EPNSugan/index\\_files/CEC2013/Definitions%20of%20CEC%2013%20benchmark%20suite%200117.pdf](http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2013/Definitions%20of%20CEC%2013%20benchmark%20suite%200117.pdf)
- [4] <http://aragorn.pb.bialystok.pl/~wkwedlo/EA6.pdf>
- [5] <http://www.mini.pw.edu.pl/~mandziuk/2012-01-12.pdf>