

## WSI – ćwiczenie 2

### Algorytmy ewolucyjne i genetyczne

Adam Wróblewski

#### **Wstęp:**

Ćwiczenie polegało na wykorzystaniu algorytmu genetycznego do znalezienia najlepszego rozwiązania zadania polegającego na lądowaniu rakiety.

Funkcja określająca jakość wyznaczonego rozwiązania ewaluowała zużycie paliwa podczas procesu, a także czy rakieta w zadanej liczbie kroków czasowych wylądowała, rozbiła się lub znajduje się w powietrzu i na tej podstawie zwracała ocenę danego rozwiązania w postaci liczby z zakresu  $(-1200; 2000)$ , gdzie im wyższa wartość, tym lepsze rozwiązanie.

Algorytm genetyczny w każdej iteracji wyznacza nową, różną od poprzedniej populację rozwiązań, ocenia je i wybiera najlepsze rozwiązanie, jakie pojawiło się w podczas działania algorytmu w zadanej liczbie iteracji  $t_{max}$ .

Implementacja algorytmu znajduje się w pliku *WSI\_Zad\_2.ipynb*.

#### **Eksperymenty i analiza wpływu hiperparametrów na działanie algorytmu:**

Eksperymenty zacząłem od znalezienia parametrów, które dają satysfakcjonujący wynik, są to:

$t_{max} = 200$	- liczba iteracji algorytmu
$pc = 0.5$	- prawdopodobieństwo krzyżowania
$pm = 0.05$	- prawdopodobieństwo mutacji
$mi = 50$	- liczba osobników w populacji

### Wpływ prawdopodobieństwa krzyżowania na działanie algorytmu:

Dla każdej wartości  $pc$  wykonałem serię 25 symulacji, obliczyłem średni wynik oraz odchylenie standardowe, a także wartość najlepszego i najgorszego wyniku.

Ponieważ wyniki były bardzo zbliżone do siebie, przeprowadziłem analizę dla 3 różnych wartości liczby iteracji:

Pozostałe parametry:  $pm = 0.05$ ,  $mi = 50$

t_max	pc	średnia	odch st	min	max
50	0	1920,60	1,65	1918	1924
50	0,001	1920,60	2,15	1917	1925
50	0,01	1919,96	1,73	1917	1923
50	0,1	1919,68	1,54	1917	1924
50	0,3	1919,88	1,70	1917	1923
50	0,5	1920,60	1,92	1917	1924
50	0,7	1919,88	1,73	1917	1924
50	0,9	1919,80	1,52	1917	1924
100	0	1921,80	1,72	1919	1925
100	0,001	1921,16	1,97	1917	1924
100	0,01	1921,04	1,75	1918	1925
100	0,1	1921,20	1,36	1919	1924
100	0,3	1922,08	2,15	1918	1928
100	0,5	1921,72	2,29	1917	1928
100	0,7	1921,28	1,43	1918	1924
100	0,9	1920,96	1,25	1918	1924
500	0	1923,76	1,53	1920	1926
500	0,001	1923,92	1,62	1920	1927
500	0,01	1923,76	1,18	1921	1926
500	0,1	1923,36	1,52	1921	1928
500	0,3	1923,36	1,47	1921	1927
500	0,5	1923,24	1,66	1921	1928
500	0,7	1923,08	1,06	1921	1926
500	0,9	1923,32	1,69	1921	1928

Otrzymane wyniki są bardzo zbliżone. Wartość prawdopodobieństwa krzyżowania zdaje się nie mieć szczególnego wpływu na rozwiązanie o ile parametr  $pc$  jest różny od zera – w przeciwnym wypadku najlepszym rozwiązaniem było by najlepsze rozwiązanie w początkowej populacji.

### Wpływ prawdopodobieństwa mutacji na działanie algorytmu:

Dla każdej wartości  $pm$  wykonałem serię 25 symulacji, obliczyłem średni wynik oraz odchylenie standardowe, a także wartość najlepszego i najgorszego wyniku.

Ponieważ wyniki były bardzo zbliżone do siebie, przeprowadziłem analizę dla 3 różnych wartości liczby iteracji, aby sprawdzić czy wyniki będą się różnić:

Pozostałe parametry:  $pc = 0.5$ ,  $mi = 50$

t_max	pm	średnia	odch st	min	max
50	0	1915,64	2,24	1911	1921
50	0,001	1915,92	2,10	1913	1921
50	0,01	1919,28	1,99	1915	1923
50	0,1	1920,28	2,03	1916	1925
50	0,3	1919,24	2,12	1916	1925
50	0,5	1918,64	1,60	1916	1923
50	0,7	1918,52	1,45	1916	1922
50	0,9	1918,44	1,75	1916	1922
100	0	1916,32	1,89	1913	1920
100	0,001	1917,16	2,07	1913	1922
100	0,01	1919,52	1,81	1917	1923
100	0,1	1920,96	1,61	1918	1926
100	0,3	1920,84	2,24	1917	1927
100	0,5	1920,08	1,55	1917	1924
100	0,7	1919,84	2,03	1917	1926
100	0,9	1919,56	1,75	1917	1923
500	0	1916,04	2,52	1910	1921
500	0,001	1918,28	2,09	1912	1922
500	0,01	1923,12	2,18	1920	1929
500	0,1	1923,32	1,49	1921	1927
500	0,3	1922,08	2,19	1919	1930
500	0,5	1922,48	1,68	1920	1926
500	0,7	1922,72	2,18	1920	1930
500	0,9	1922,2	1,44	1920	1925

Najlepszą wartością prawdopodobieństwa mutacji zdaje się być 0,1. Różnica wyniku między wartościami mutacji zerowymi lub bliskimi zero jest zauważalna ( ok 6 punktów ), natomiast wyniki algorytmu dla wartości prawdopodobieństwa mutacji większej od 0.1 są nieco gorsze.

### Wpływ liczby iteracji na działanie algorytmu:

Dla każdej wartości  $t_{max}$  wykonałem serię 25 symulacji, obliczyłem średni wynik oraz odchylenie standardowe, a także wartość najlepszego i najgorszego wyniku.

Pozostałe parametry:  $pc = 0.5$ ,  $pm = 0.05$ ,  $mi = 50$

t_max	średnia	odch st	min	max
10	1917,24	2,16	1915	1925
100	1921,24	1,73	1917	1924
500	1923,56	1,50	1922	1928
1000	1923,44	1,10	1921	1925
10000	1926	0,94	1925	1928

Liczba iteracji ma wpływ na osiągnięty przez algorytm wynik. Im więcej iteracji tym średnia wartość wyznaczonego rozwiązania jest lepsza. Ma to sens gdyż zwiększając liczbę iteracji zwiększamy całkowitą liczbę osobników którzy zostaną wygenerowani podczas działania algorytmu, a tym samym szansę na wygenerowanie osobnika, którego wartość funkcji celu jest wysoka. Odchylenie standardowe zmniejsza się wraz ze wzrostem iteracji, co świadczy o większej spójności wyników. Niestety przy dużej liczbie iteracji czas potrzebny na rozwiązanie jest bardzo długi.

### Wpływ rozmiaru populacji na działanie algorytmu:

Dla każdej wartości  $mi$  wykonałem serię 25 symulacji, obliczyłem średni wynik oraz odchylenie standardowe, a także wartość najlepszego i najgorszego wyniku.

Pozostałe parametry:  $t_{max} = 200$ ,  $pc = 0.5$ ,  $pm = 0.05$

mi	średnia	odch st	min	max
10	1919,60	1,65	1916	1923
50	1922,24	1,80	1920	1927
100	1923,08	2,08	1920	1928
200	1924,40	1,44	1922	1927
500	1925,16	1,43	1923	1928
1000	1925,48	1,27	1923	1930

Rozmiar populacji wpływa na rezultat działania algorytmu genetycznego. Im większa populacja tym lepszy wynik. Dzieje się tak ponieważ zwiększając populację zwiększamy całkowitą liczbę osobników, która zostanie wygenerowana podczas działania algorytmu, a tym samym prawdopodobieństwo znalezienia dobrego osobnika. Przy dużych populacjach odchylenie standardowe jest mniejsze, co znaczy o spójności wyników.

## Wnioski ogólne:

Wynik działania algorytmu genetycznego zależy od kilku parametrów, które w mniejszym bądź większym stopniu wpływają na otrzymane rozwiązanie. Parametrami które zdają się mieć największe znaczenie jest liczba iteracji  $t_{max}$  oraz rozmiar populacji  $mi$ . Niestety duże wartości tych parametrów znacząco wydłużają czas potrzebny na otrzymanie rozwiązania. Prawdopodobieństwo mutacji ma mniejszy wpływ na działanie algorytmu, parametr ten nie może być ani zbyt duży, ani zbyt mały, z przeprowadzonej analizy wynika że powinien być on równy 0.01, niestety możliwe jest że w zależności od specyfiki zadania  $pm$  będzie inne.

Ostatnim hiperparametrem jest prawdopodobieństwo krzyżowania, niestety nie byłem w stanie stwierdzić wpływu tego parametru na rezultat działania algorytmu.

Ważne jest aby  $pm$  i  $pc$  nie były jednocześnie równe 0, w takim przypadku rozwiązaniem końcowym będzie najlepsze rozwiązanie wygenerowane podczas losowej inicjalizacji populacji początkowej.