

# Raport z laboratorium 2

Marcin Sikorski

## Zadanie:

Zaimplementować klasyczny algorytm ewolucyjny bez krzyżowania, z selekcją turniejową i sukcesją generacyjną.

W raporcie wskazać jak zmiana liczby osobników w populacji wpływa na jakość uzyskanych rozwiązań przy ograniczonym budżecie.

Warto również opisać zachowanie algorytmu dla różnych rodzajów danych wejściowych oraz wpływ zmiany parametrów.

## Opis rozwiązania:

Kod jest napisany na podstawie pseudokodu.

Osobnik jest klasą, jego genotyp to lista wartości boolean reprezentujących wybrane wierzchołki grafu.

Funkcja celu ma koszt, którym jest ilość wybranych wierzchołków oraz karę, czyli 100 razy ilość niepokrytych krawędzi, niższy wynik jest lepszy.

Selekcja turniejowa najpierw losuje jednego z osobników, usuwa z listy, żeby uniknąć powtórzenia i dodaje go z powrotem po wylosowaniu drugiego. Do następnej generacji przechodzi osobnik z niższą rangą. W programie założeniem są turnieje 2-osobnikowe.

Mutacja ma najpierw prawdopodobieństwo wystąpienia jakiejkolwiek mutacji, następnie każdy bit genotypu ma oddzielne prawdopodobieństwo mutacji.

Funkcja oceniająca sortuje osobniki od najniższego wyniku funkcji do najwyższego.

## Przykład działania algorytmu:

Populacja = 500

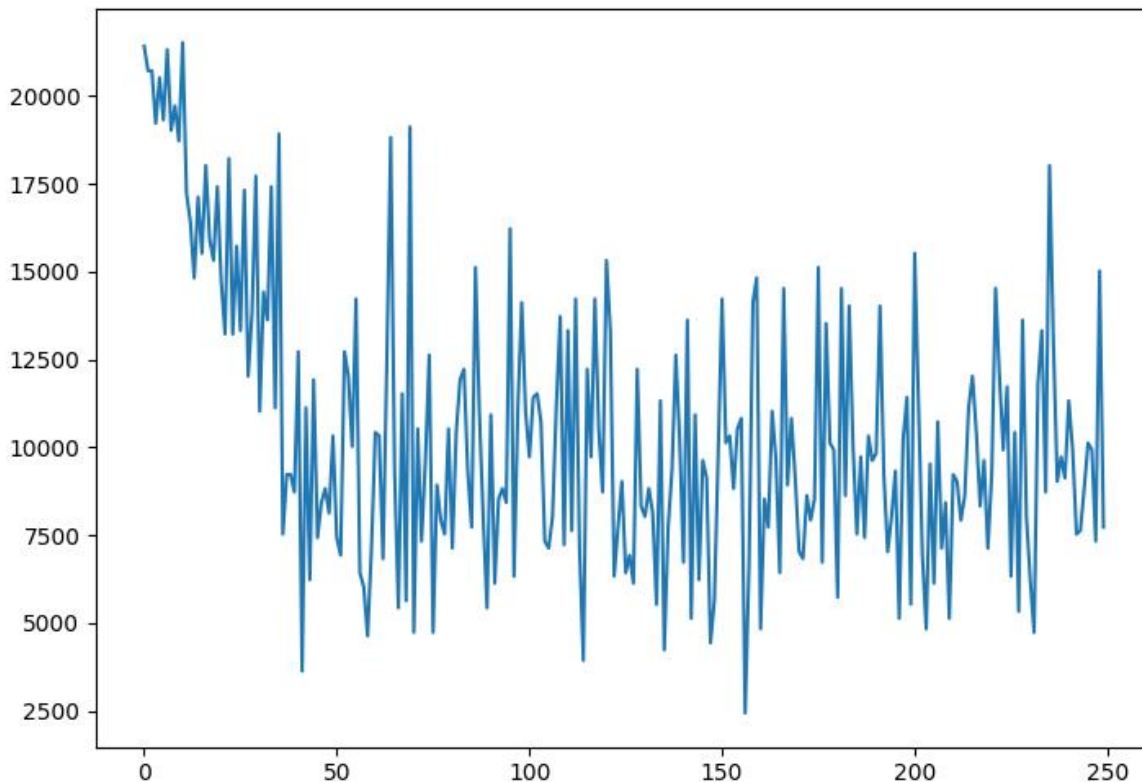
Liczba generacji = 250

Prawdopodobieństwo mutacji = 0,25

Prawdopodobieństwo zmiany bitu podczas mutacji = 0,1

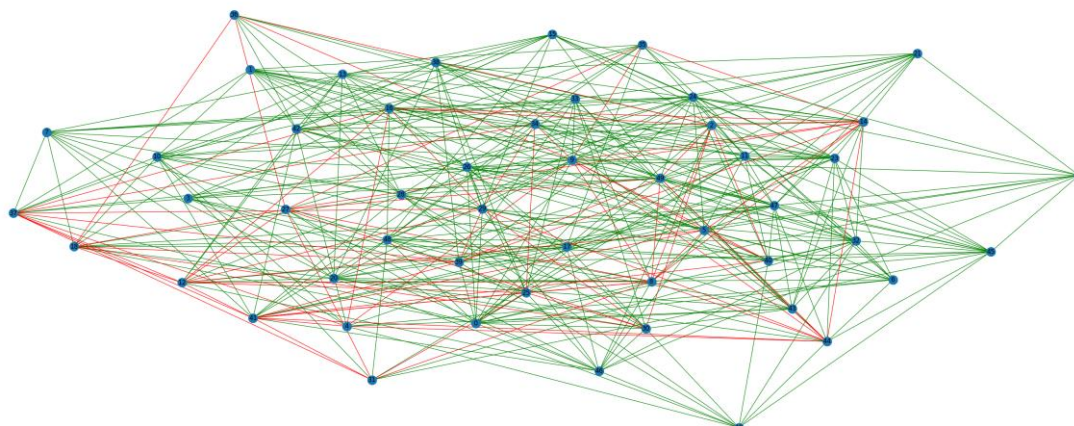
Wykres najlepszego wyniku w każdej generacji:

Niższy wynik jest lepszy.



Najlepsze rozwiązanie w ostatniej generacji:

Zielone krawędzie są oświetlone, czerwone nie.



## Wpływ ilości osobników na jakość rozwiązania:

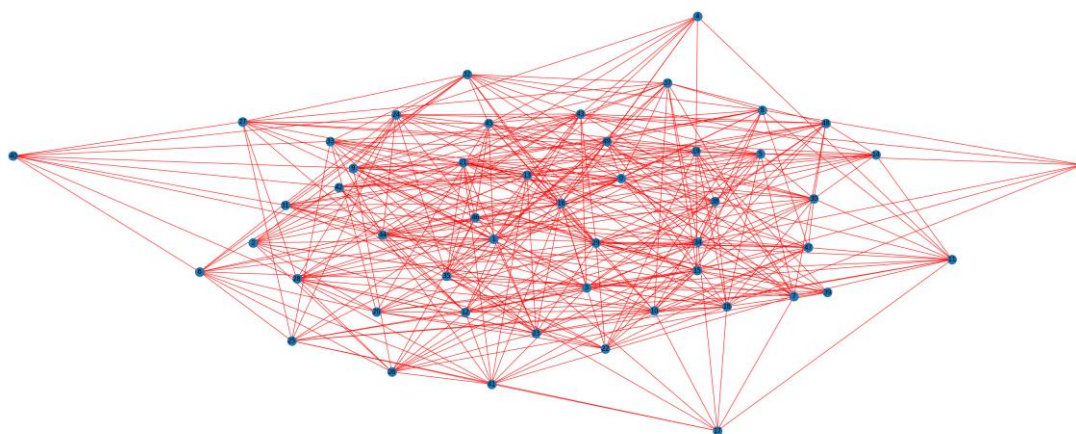
Prawdopodobieństwo mutacji = 0,25

Prawdopodobieństwo zmiany bitu = 0,1

Liczba generacji = 50

Liczba prób do uśrednienia = 10

Używany graf losowy z usuniętymi 70% krawędzi:



Liczba osobników	Maksymalny wynik	Minimalny wynik	Średnia	Odchylenie standardowe	Średni czas próby
100	13421	6830	9315,7	2291,75	2,40s
500	13818	3132	7856,0	3474,67	10,95s
2000	12618	5830	7955,4	1907,34	40,12s

Teraz te same parametry i graf, ale tylko 10 generacji.

Liczba osobników	Maksymalny wynik	Minimalny wynik	Średnia	Odchylenie standardowe	Średni czas próby
100	15518	10024	12620,4	1753,7	0,42s
500	19016	11822	15628,8	1718,04	2,23s
2000	20514	16019	18486,0	1447,67	9,49s

## Wpływ parametrów na jakość rozwiązania:

Prawdopodobieństwo mutacji:

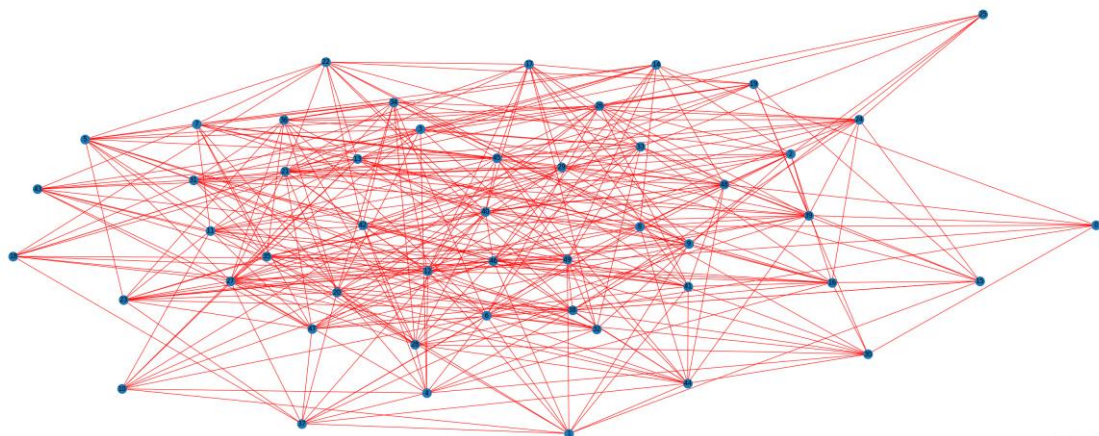
Wielkość populacji = 250

Szansa na zmianę bitu podczas mutacji = 0,1

Liczba generacji = 50

Liczba prób do uśrednienia = 10

Nowy graf losowy z usuniętymi 70% krawędzi:



Szansa mutacji	Maksymalny wynik	Minimalny wynik	Średnia	Odchylenie standardowe	Średni czas próby
0,1	14119	6130	8825,6	2769,96	5,68s
0,25	12521	5132	9635,0	2557,31	5,75s
0,75	11222	6130	8415,1	1697,15	5,05s

Prawdopodobieństwo zmiany bitu:

Wielkość populacji = 250

Szansa mutacji = 0,5

Liczba generacji = 50

Liczba prób do uśrednienia = 10

Ten sam graf

Szansa zmiany bitu	Maksymalny wynik	Minimalny wynik	Średnia	Odchylenie standardowe	Średni czas próby
0,1	15418	3233	8855,9	2925,29	5,67s
0,25	15918	5529	9574,2	3351,09	5,39s
0,75	11222	6130	10784,3	4053,56	5,83s

## Wnioski:

- Algorytm po pewnej liczbie generacji nie osiąga lepszych wyników
- Dla dużej liczby generacji lepsza jest większa populacja
- Jeśli zasoby są ograniczone i liczba generacji jest mała, mniejsza populacja daje lepszy wynik
- Większe prawdopodobieństwo mutacji powoduje lepsze wyniki po dużej ilości generacji
- Duże prawdopodobieństwo zmiany bitu powoduje gorsze wyniki, z dużym odchyleniem standardowym