

## ALGORYTM GENETYCZNY – VERTEX COVER PROBLEM

Do rozwiązania problemu został użyty algorytm genetyczny z selekcją turniejową 2 osobnikową, losową mutacją i bez krzyżowania.

Algorytm na początku tworzy mapę połączeń między “latarniami” oraz populację osobników, z których każdy jest listą 0 i 1 (0 – latarnia zgaszona, 1- zapalona).

Następnie, każdy z osobników zostaje porównany z bazą wymaganych połączeń i wstępnie oceniany. Wtedy zaczyna się pętla, która powtarza proces oceny, dokonując dodatkowo mutacji z określonym prawdopodobieństwem w osobnikach, które wygrały porównania z losowo dobranym “przeciwnikiem”.

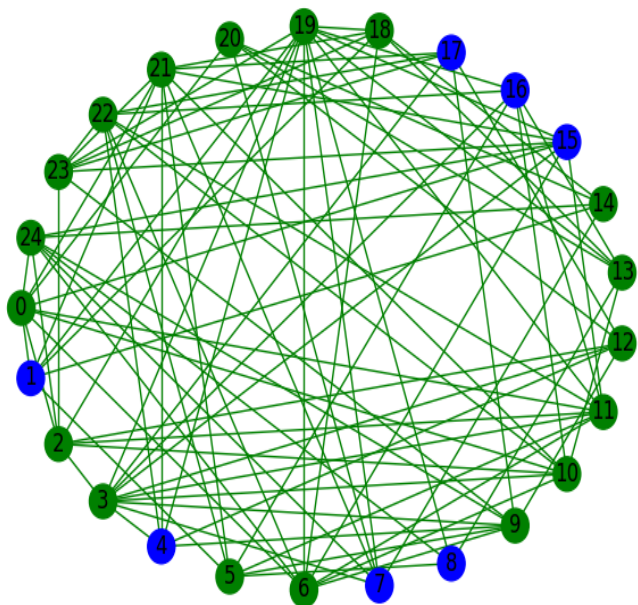
Po dobraniu parametrów, przy kilku próbach na 10 “latarniach” i 100 osobnikach oraz tej samej bazie połączeń:

Średni czas: 0,149 sekundy

Najlepszy wynik: 7 zgaszonych latarni na 20

Najgorszy wynik: 6 zgaszonych latarni na 20

Parametry użyte do opisanego testu znajdują się na prawym obrazku poniżej.



```
# *****number of nodes (problem size) *****
s = 10
# *****number of individuals in population *****
n = 100
# *****number of iterations *****
iterations = 40
# *****filling the graph *****
fill = 0.6
# *****filling start population with ones *****
fill_pop = 0.3
# *****probability of mutation *****
mut = 0.1
# *****loss multiplier *****
loss_multiplier = 100
# *****probability of better solution winning *****
good_win_prob = 1
```

Pierwszy zrzut ekranu przedstawia wizualizację efektów pracy algorytmu.

Niebieski punkt – latarnia zgaszona, zielony – latarnia zapalona.

Wszystkie połączenia są zielone, więc ścieżki są oświetlone zgodnie z założeniem.

Jeśli któreś połączenie byłoby czerwone, oznaczałoby to że jest niepokryte.

Biblioteki wykorzystane w projekcie:

networkx, matplotlib.pyplot- do tworzenia grafów

numpy - do przetwarzania macierzy

time – do pomiaru czasu wykonania

datetime – do tworzenia znaczników czasu wytworzenia matryc połączeń