

WSI Lab 2 – algorytmy genetyczne i ewolucyjne

Zadanie

Zaimplementowanie algorytmu ewolucyjnego bez krzyżowania, z selekcją turniejową i sukcesją generacyjną.

Uruchomienie programu

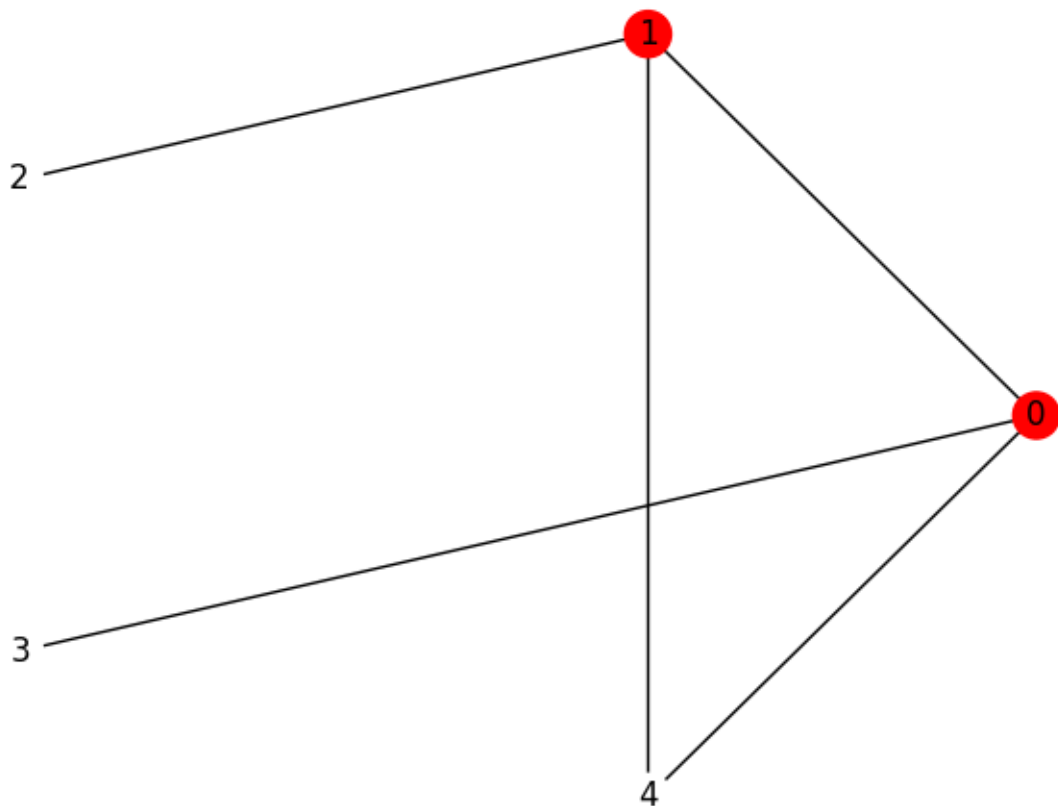
```
python3.9 evolution.py [--fullness FULLNESS] [--mutation PROBABILITY] [--iterations ITERATIONS] [--tournament SIZE] [--graph] vertices covered population
```

Żeby dowiedzieć się więcej na temat argumentów należy użyć flagi "-h".

Wyniki

Wynik działania algorytmu dla parametrów:

- Rozmiar grafu – 5
- Graf pełny z usuniętymi 50% krawędzi
- Rozmiar turnieju – 2
- Prawdopodobieństwo mutacji – 5%
- Rozmiar populacji - 50



Gdzie czerwone wierzchołki są pokryte.

Badane wpływu parametrów

Domyślne ustawienia:

- Rozmiar grafu – 25
- Początkowa liczba pokrytych wierzchołków - 18
- Graf pełny z usuniętymi 50% krawędzi
- Rozmiar turnieju – 2
- Prawdopodobieństwo mutacji – 5%

Wpływ rozmiaru populacji na rezultat działania algorytmu (końcową liczbę wierzchołków niezbędnych do pokycia wszystkich krawędzi).

Wynik po 15 uruchomieniach algorytmu:

populacja	min	średnia	max	odchylenie
25	20	20.93	21	0.258
50	20	20.6	21	0.507
100	20	20.53	21	0.516
200	20	20.27	21	0.458
300	20	20.27	21	0.458

Oraz wpływ rozmiaru populacji na czas wykonania programu:

populacja	min[s]	średnia[s]	max[s]	odchylenie[s]
25	2.35	3.24	4.34	0.433
50	4.23	5.88	7.25	0.771
100	8.56	11.58	14.22	1.665
200	15.69	22.3	26.17	3.097
300	23.22	33.95	41.36	5.160

Im większy rozmiar populacji, tym większa szansa, że jeden z osobników już jest lub jest bardzo bliski rozwiązania. Ma to jednak duży wpływ na czas wykonania programu.

Jak widać z tabeli minimalny czas wykonania jest prawie wprost proporcjonalny do rozmiaru populacji, wydaje się, że wiąże się to z tym, że w przypadkach dla minimalnego czasu wykonania programu rozwiązanie już było jednym z osobników populacji, więc dłuższa część czasu to sortowanie wszystkich osobników oraz sprawdzenie czy nie są rozwiązaniami. Natomiast zgodnie z faktem, że im większy rozmiar populacji, tym większa szansa, że jeden z osobników już jest rozwiązaniem stosunek średniego czasu do rozmiaru populacji jest lepszy niż wprost proporcjonalny.

Rozmiar turnieju = 4.

Wyniki dla domyślnych ustawień oraz rozmiaru turnieju = 4:

populacja	min	średnia	max	odchylenie
25	20	20.6	21	0.507
50	20	20.73	21	0.458
100	20	20.73	21	0.458
200	20	20.73	21	0.458
300	20	20.6	21	0.507

Oraz wpływ rozmiaru populacji na czas wykonania programu:

populacja	min[s]	średnia[s]	max[s]	odchylenie[s]
25	3.2	4.71	5.33	0.597
50	6.7	9.22	10.08	1.08
100	12.7	18.17	20.58	2.04
200	25.2	35.66	41.45	5.7
300	37.58	54.27	64.94	6.7

Po 15 uruchomieniach okazało się, że dla większego rozmiaru turnieju dostano gorsze wyniki oraz znacznie dłuższy czas wykonania programu.

Im większy rozmiar turnieju, tym mniejsze prawdopodobieństwo, że słabe osobniki zostaną wybrane, co zmniejsza różnorodność populacji po reprodukcji. Natomiast im mniejszy rozmiar turnieju, tym większe prawdopodobieństwo, że wszystkie osobniki populacji zostaną wybrane, co zwiększa różnorodność.

Prawdopodobieństwo mutacji = 15%

Wyniki dla domyślnych ustawień oraz prawdopodobieństwa mutacji = 15% po 15 uruchomieniach:

populacja	min	średnia	max	odchylenie
25	20	20.6	21	0.507
50	20	20.87	21	0.35
100	20	20.53	21	0.516
200	20	20.67	21	0.487
300	20	20.67	21	0.487

Oraz wpływ rozmiaru populacji na czas wykonania programu:

populacja	min[s]	średnia[s]	max[s]	odchylenie[s]
25	2.16	2.89	3.55	0.43
50	4.95	6.02	6.66	0.41
100	7.82	10.93	12.77	1.61
200	17.98	22.57	25.41	2.53
300	23.94	33.1	37.02	4.92

Jak widzimy, wynik okazał się podobny do wyniku dla prawdopodobieństwa mutacji = 5%.

Ogólne im większe prawdopodobieństwo mutacji, tym większa różnorodność populacji. Natomiast zbyt duże prawdopodobieństwo mutacji zapobiega konwergencji populacji do jakiegokolwiek optymalnego rozwiązania.