

## Projektowanie efektowych algorytmów. Projekt 3 Algorytm Genetyczny

### 1. Wstęp teoretyczny

Algorytm genetyczny to rodzaj heurystyki (czyli algorytm który nie gwarantuje znalezienia rozwiązania optymalnego), którego działanie opiera się na przeszukiwaniu przestrzeni alternatywnych rozwiązań problemu w celu wyszukiwania rozwiązań lepszych od poprzednich. Jak sama nazwa wskazuje algorytm przypomina zjawisko ewolucji biologicznej i opiera się na działaniach genetycznych takich jak: selekcja, krzyżowanie czy mutacja.

#### 1.2 Definicje operatorów genetycznych:

**selekcja** - polega na wyborze z bieżącej populacji osobników, których materiał genetyczny zostanie poddany operacji krzyżowania oraz mutacji i przekazany osobnikom potomnym (kolejna populacja). Wybór następuje na podstawie określonej metody selekcji. Osobniki oceniane są na podstawie tzw. funkcji przystosowania/dopasowania (ang. fitness function - ff). Selekcja powinna promować osobniki najlepiej przystosowane - o najwyższej wartości ff

**krzyżowanie** (ang. crossover) - polega na wymianie materiału genetycznego pomiędzy losowo wybranymi (podczas selekcji) parami osobników. W wyniku krzyżowania powstają nowe osobniki, które mogą wejść w skład nowej populacji (kolejnego pokolenia). Osobniki powstałe po krzyżowaniu powinny być - i są, jeśli metoda selekcji jest właściwa - lepiej przystosowane (wyższa wartość ff) od swoich rodziców. Zachodzi z prawdopodobieństwem  $p_c$ .

**mutacja** - polega na zamianie wartości losowo wybranego genu (cechy osobnika). Celem użycia operatora mutacji jest zapewnienie zmienności chromosomów. W przypadku wykorzystania AG do poszukiwania rozwiązania np. problemów kombinatorycznych stwarza możliwość wyjścia z optimum lokalnych i lub zwiększenia intensyfikacji przeszukiwania. Zachodzi z prawdopodobieństwem  $p_m$ .

### 2. Opis działania algorytmu

- 2.1. Losowana jest pewna populacja początkowa.
- 2.2. Populacja poddawana jest ocenie (**selekcja**). Najlepiej przystosowane osobniki biorą udział w procesie reprodukcji.
- 2.3. Genotypy wybranych osobników poddawane są operatorom ewolucyjnym:
  - 2.3.1. są ze sobą kojarzone poprzez złączanie genotypów rodziców (**krzyżowanie**),
  - 2.3.2. przeprowadzana jest mutacja, czyli wprowadzenie drobnych losowych zmian.
- 2.4. Rodzi się drugie (kolejne) pokolenie. Aby utrzymać stałą liczbę osobników w populacji te najlepsze (według funkcji oceniającej fenotyp) są powielane, a najgorsze usuwane. Jeżeli nie znaleziono dostatecznie dobrego rozwiązania,

algorytm powraca do kroku drugiego. W przeciwnym wypadku wybieramy najlepszego osobnika z populacji - jego genotyp to uzyskany wynik.

W algorytmie zostały do krzyżowania został użyty algorytm OX. Natomiast populacja została dobrana metodą turniejową.

### 3. Badania dla różnych instancji

#### 3.1 Badania dla pliku gr17.tsp - dla 17 miast w grafie symetrycznym

Czas operacji [ms]	Liczba iteracji	Wielkość populacji	Liczba par dzieci	Prawdopodobieństwo mutacji	Obliczona droga	Błąd względny
8	5	5	10	0.1	3712	78,033573141
23	50	5	10	0.1	2601	24,748201438
62	50	5	100	0.1	2202	5,6115107913
70	50	50	100	0.1	2202	5,6115107913
317	50	50	1000	0.1	2170	4,0767386091
380	50	500	1000	0.1	2153	3,2613908872
2271	500	500	1000	0.1	2085	0
1930	500	500	1000	0.01	2085	0

Z powyższej tabeli wynika, że czynnikami wpływającymi najbardziej na dokładność wyszukiwanego wyniku jest wielkość populacji i liczba par dzieci. Współczynnik mutacji jak widać dla małych instancji nie wpływa znacząco. Czas operacji jest uzależniony od liczby iteracji, wielkości populacji i liczby par dzieci - czym większe tym czas większy. Warto jednak wspomnieć o dużej losowości otrzymanych wyników, gdyż otrzymana wartość zależy w dużej mierze od początkowej populacji, która jest dobierana losowo.

Dla porównania w programowaniu dynamicznym czas rozwiązania wynosił 912ms, lecz jego rozwiązanie gwarantowało znalezienia rozwiązania optymalnego, natomiast dla tabu search najlepszy wynik wynosił także 2085 jednak algorytm wykonał się w czasie 98ms, czyli znacznie szybciej.

#### 3.2 Badania dla średniej instancji 48 miast grafu symetrycznego (plik gr48.tsp)

Czas operacji [ms]	Liczba iteracji	Wielkość populacji	Liczba par dzieci	Prawdopodobieństwo mutacji	Obliczona droga	Błąd względny
12	5	5	10	0.1	18760	271,77962742
34	50	5	10	0.1	12863	154,91478398
164	50	5	100	0.1	8524	68,925881886
160	50	50	100	0.1	8533	69,104240982
808	50	50	1000	0.1	6856	35,869996036
1375	50	500	1000	0.1	5805	15,041617122

Czas operacji [ms]	Liczba iteracji	Wielkość populacji	Liczba par dzieci	Prawdopodobieństwo mutacji	Obliczona droga	Błąd względny
5957	500	500	1000	0.1	5311	5,2516845025
15965	500	500	1000	0.01	5606	11,097899326

Jak widać na powyższej tabeli najlepiej sprawdza się populacja wielkości 500 i liczba dzieci np. 1000. Bardzo ważnym czynnikiem wpływającym na zmniejszenie drogi jest liczba iteracji, która oczywiście wydłuża czas wykonywania algorytmu. Jak widać, dla powyższego miasta zwiększone prawdopodobieństwo mutacji do 10% wpływa pozytywnie na otrzymany wynik.

W porównaniu do tabu search otrzymaliśmy jednak gorsze wyniki, gdyż najkrótszą obliczoną drogą w tabu search była droga o długości 5351

### 3.3. Badania dla dużej instancji 262 miast grafu symterycznego(plik gil262.tsp)

Czas operacji [ms]	Liczba iteracji	Wielkość populacji	Liczba par dzieci	Prawdopodobieństwo mutacji	Obliczona droga	Błąd względny
86	5	5	10	0.1	24661	937,04793944
74	50	5	10	0.1	22159	831,83347350
1448	50	5	100	0.1	18757	688,77207737
6679	50	50	100	0.1	14177	496,17325483
8058	50	100	100	0.1	17087	618,54499579
10601	200	50	100	0.1	11321	376,07232968
15933	500	50	100	0.1	8855	272,37174095
16668	500	50	100	0.01	9706	308,15811606
54499	500	100	400	0.1	8369	251,93439865
150191	2000	100	400	0.1	6932	191,50546677
229119	2000	200	400	0.1	5573	134,35660218

Z powyższej tabeli ponownie wynika że największy wpływ na znajdowanie drogi o niższym koszcie ma liczba iteracji. Najlepiej się sprawdziła liczba par dzieci rzędu 400 i wielkość populacji rzędu 200. Dla powyższego miasta dobrze także zadziałało zwiększenie prawdopodobieństwa mutacji. W porównaniu do tabu search także otrzymaliśmy gorsze wyniki gdyż tam najlepszym wynikiem okazał się wynik 3718.

### 3.4 Badania dla małych instancji 17 miast w grafie asymetrycznym (br17.atsp)

Czas operacji [ms]	Liczba iteracji	Wielkość populacji	Liczba par dzieci	Prawdopodobieństwo mutacji	Obliczona droga	Błąd względny
7	5	5	10	0.1	115	194,87179487
19	50	5	10	0.1	58	48,717948717
62	50	5	100	0.1	51	30,769230769
46	50	50	100	0.1	40	2,5641025641
376	50	500	1000	0.1	39	0

Jak widać tendencja dla małych instancji się utrzymuje także dla grafów asymetrycznych. Ponownie liczba iteracji, a także wielkość populacji i liczba par dzieci wpływają w największym stopniu na zmniejszenie obliczonej drogi, jednak zwiększa ona czas wykonania algorytmu.

Tabu search w trakcie 98 ms i programowanie dynamiczne w ciągu 930 ms także znalazły optymalną drogę dla powyższego grafu. Jak widać więc algorytm genetyczny jest wolniejszy od tabu search, natomiast programowanie dynamiczne wykonuje się najdłużej jednak oferuje gwarancję optymalności otrzymanego wyniku.

### 3.5 Badanie dla średnich instancji 47 miast w grafie asymetrycznym (ftv47.atsp)

Czas operacji [ms]	Liczba iteracji	Wielkość populacji	Liczba par dzieci	Prawdopodobieństwo mutacji	Obliczona droga	Błąd względny
9	5	5	10	0.1	5246	195,38288288
49	50	5	10	0.1	5115	188,00675675
144	50	5	100	0.1	3195	79,898648648
187	50	50	100	0.1	2838	59,797297297
901	50	50	1000	0.1	2492	40,315315315
341	50	500	1000	0.1	2123	19,538288288
3492	200	500	1000	0.1	2001	12,668918918
2771	200	500	1000	0.01	1974	11,148648648
12965	200	2000	4000	0.01	1848	4,0540540540
23097	200	4000	8000	0.01	1834	3,2657657657
26446	200	4000	8000	0.1	1825	2,7590090090

Jak widać na powyższej tabeli dla średnich miast ma największe znaczenie wielkość populacji i ilość dzieci. Liczba iteracji powinna wynosić około 200 iteracji gdyż dla takiej wielkości miast w większej liczbie iteracji nie następuje poprawa. W porównaniu do tabu search otrzymaliśmy lepsze wyniki gdyż najlepszym wynikiem

osiągniętym przez tabu search była droga o długości 2002 odnaleziona w czasie 7097ms. Widać także że wielkość populacji powinna być około 2 razy mniejsza od liczby par dzieci.

### 3.6 Badanie dla dużych instancji (rbg323.atsp)

Czas operacji [ms]	Liczba iteracji	Wielkość populacji	Liczba par dzieci	Prawdopodobieństwo mutacji	Obliczona droga	Błąd względny
171	5	5	10	0.1	5746	333,33333333
285	50	5	10	0.1	5085	283,48416289
2215	50	5	100	0.1	4200	216,74208144
10291	50	50	100	0.1	3945	197,51131221
24003	200	50	100	0.1	2939	121,64404223
30214	500	50	100	0.1	2530	90,799396681
31293	500	50	100	0.01	2852	115,08295625
261287	500	200	400	0.1	2033	53,318250377

Jak wynika z powyższej tabeli najważniejszym czynnikiem wpływającym na dokładność wyniku jest wielkość populacji i liczba dzieci. Ważna jest aby ilość par dzieci było 2 razy większa niż wielkość populacji. Ilość iteracji także powinna być większa niż dla mniejszych miast. Dla powyższego grafu wyniki algorytmu genetycznego są gorsze od tabu search gdyż najlepszy wynik tabu search wynosi 1602.

## 4.Podsumowanie i wnioski

Podsumowując, najważniejszym czynnikiem wpływającym na dokładność wykonanego algorytmu jest wielkość populacji i liczba par dzieci. Jednak zwiększenie tych parametrów wpływa na wydłużenie czasu wykonywania algorytmu. Jeśli chodzi o populację, ta powinna być około 4 krotnie mniejsza od liczby par dzieci. Dzięki temu w kolejnych pokoleniach wyselekcjonowani są tylko ci rodzice, których droga jest najmniejsza i rodziców z dużą drogą pozostaje stosunkowo niewiele. Mutacja rzędu 10% także wpływa pozytywnie na otrzymany wynik. Liczba iteracji powinna natomiast zostać dostosowana do liczby miast - czym więcej miast tym większa powinna być liczba iteracji. Warto jednak wspomnieć o dużej losowości otrzymanego wyniku, gdyż otrzymany wynik zależy w dużej mierze od populacji początkowej która jest dobierana losowo.

Porównując natomiast wszystkie trzy algorytmy, które zostały wykonane na projekcie, (tabu search, programowanie dynamiczne i algorytm genetyczny) programowanie dynamiczne gwarantuje zawsze poprawność otrzymanego wyniku, natomiast tabu search oferuje największą szybkość w stosunku do jakości

otrzymanego wyniku. Niestety spośród tych trzech algorytmów algorytm genetyczny wypadł najgorzej.