Wrocław, 12.01.2020

Adrian Śnieżek
235984
Śr 17.05

Prowadzący dr inż. Jarosław Mierzwa

Sprawozdanie
Projektowanie efektywnych algorytmów
Projekt III

1. Wstęp teoretyczny

Algorytm genetyczny jest algorytmem stosowanym w celu znalezienia optymalnego rozwiązania danego problemu – są to algorytmy niedokładne.

Algorytm genetyczny przypomina zjawisko ewolucji biologicznej i jest oparty na biologicznych pojęciach ewolucyjnych. Został stworzony przez Johna Henrego Hollanda, który czerpał inspiracje z nauk biologii.

Algorytm polega na wyborze osobników do stworzenia populacji. Następnie dzięki krzyżowaniu dochodzi do reprodukcji i generowane jest kolejne pokolenie. Przeprowadzana jest również mutacja, która ma na celu wprowadzać losowe drobne zmiany w genotypie osobników. Do zachowania stałej wielkości populacji należy zastosować sposób wyboru osobników do następnego pokolenia. Algorytm zostaje przeprowadzony aż do osiągnięcia kryterium stopu.

2. Algorytm genetyczny a problem komiwojażera

W celu znalezienia jak najlepszej drogi i przeprowadzenia algorytmu należało przyjąć:

a) Drogę jako osobnika
b) Zbiór różnych dróg jako populację
c) Tworzenie potomka jako krzyżowanie dwóch dróg
d) Mutacje czyli zmianę kolejności miast w drodze

Napisany algorytm tworzy pierwszą populację o domyślnej wielkości 20 osobników, które zostają wygenerowane metodą zachłanną. Aby zapewnić różnorodność każdy osobnik rozpoczyna się od innego wierzchołka – miasta.

Posiadając populację dochodzi do wyboru par osobników. Algorytm przechodzi przez kombinację każdy z każdym dokładnie dwa razy. Krzyżowanie metodą order crossover zachodzi z określonym prawdopodobieństwem ustawionym domyślnie na wartość równą 0,8.

Po wygenerowaniu nowych osobników dochodzi do ustalenia następnej populacji. W tym celu odrzuca się najgorszych osobników populacji pierwotnej i dodaje najlepszych z nowego pokolenia zachowując stałą wielkość populacji.

Ostatnim krokiem jest przeprowadzenie mutacji. Mutowanie genotypów osobników zachodzi metodą inwersji czyli wybranie losowego fragmentu genotypu oraz odwrócenie go. Mutacja zachodzi z niskim prawdopodobieństwem i domyślnie została ustawiona na wartość 0,01.

3. Krzyżowanie

Krzyżowanie osobników przeprowadzane jest metodą order crossover polegającej na:

a) Wybraniu 2 osobników jako rodziców: P_1 (1, 2, 3, 4, 5, 6, 7, 8, 9) oraz P_2 (9, 3, 7, 8, 2, 6, 5, 1, 4)
b) Skopiowanie segmentu genotypu rodzica P_1 np (1, 2, 3, 4, 5, 6, 7, 8, 9)
c) Przeniesienie fragmentu do potomka na odpowiadającą pozycję - (-, -, -, 4, 5, 6, 7, -, -)
d) Ustawienie się na pozycję za ostatnią skopiowaną pozycją, w tym przypadku za liczbą „7”
e) Kopiowanie kolejnych pozycji z rodzica P_2 pomijając pozycje występujące już w potomku, w tym przykładnie: 1-4-9-3-7-8-2-6-5. Kolorem czerwonym oznaczono pozycje pominięte.
f) Otrzymany zostaje potomek - (3, 8, 2, 4, 5, 6, 7, 1, 9)

4. Mutacja

Mutacja osobników przeprowadzana jest metodą inwersji polegającej na:

a) Wybranie losowego fragmentu genotypu osobnika np. (1, 2, 3, 4, 5, 6, 7, 8, 9)
b) Odwrócenie fragmentu na przeciwny 4, 5, 6, 7 -> 7, 6, 5, 4
c) Nadpisanie osobnika o odwrócony fragment genotypu i otrzymanie zmutowanego osobnika (1, 2, 3, 7, 6, 5, 4, 8, 9)

5. Opis działania programu

W programie zaimplementowałem klasę sterująca „menu”, która służy za interfejs użytkownika. Obiekt przy utworzeniu wyświetla opcje jakie użytkownik może wybrać. W zależności od wyborów wywoływane są poszczególne metody. Dostępne opcje to:

- wczytanie danych - następuje wyświetlenie plików z rozszerzeniem „.atsp” znajdujących się w folderze z danymi wejściowymi, użytkownik wpisuje nazwę pliku i wybrane dane zostają wczytane

- ustawienie czasu – użytkownik decyduje przez jaką ilość sekund algorytm będzie wykonywany

- ustawienie współczynnika krzyżowania – użytkownik decyduje o prawdopodobieństwie zajścia krzyżowania między osobnikami

- ustawienie współczynnika mutacji – użytkownik decyduje o prawdopodobieństwie zajścia mutacji osobników

- wyświetlenie aktualnych parametrów – następuje wyświetlenie ustawionego czasu i współczynników prawdopodobieństwa. Jeżeli wartości nie zostaną ustalone przez użytkownika czas domyślny wynosi 60 sekund, współczynnik krzyżowania 0,8 a współczynnik mutacji 0,01

- wyświetlenie aktualnych danych – następuje wyświetlenie macierzy wag przejść między wierzchołkami

- algorytm genetyczny– opcja wywołująca algorytm genetyczny dla określonych lub domyślnych parametrów

- algorytm genetyczny (10 times) – wywołanie serii 10 powtórzeń algorytmu genetycznego

- seria testów – wykonanie 10-ciu wywołań algorytmu genetycznego (plik mały – 20 minuty, plik średni – 4 minuty, plik duży – 6 minut)

- koniec – wyjście z programu

Klasa „menu” korzysta jednakże z drugiej klasy – „controller” , która jest odpowiedzialna za przeprowadzenie algorytmów na podstawie danych przekazanych przez klasę „menu”.

Klasa „controller” posiada metody takie jak:

- krzyżowanie dwóch osobników

- mutację osobnika

- znajdowanie najlepszego osobnika w populacji

- znajdowanie najgorszego osobnika w populacji

- inicjalizację drogi metodą zachłanną

- generację populacji pierwotnej

- kalkulacje kosztu drogi

- przeprowadzenie algorytmu genetycznego

6. Przeprowadzenie doświadczeń i środowisko pracy

Wykonywane doświadczenia zostały przeprowadzone na komputerze stacjonarnym o podzespołach:

- płyta główna ASUS Prime b350-plus

- procesor AMD Ryzen 7, 8 rdzeni, 3.65 GHz

- pamięć RAM CORSAIR Vengeance Lpx 16 GB 3000 MHz DDR4

Z wykorzystaniem system operacyjnego Windows 10 Education, pracując z językiem Python w środowisku programistycznym JetBrains PyCharm.

Założenia:

- ustalenie stałego czasu wykonywania algorytmu w zależności od rozmiaru pliku:
  a) plik mały – 2 minuty
  b) plik średni – 4 minuty
  c) plik duży – 6 minut

- dla każdego z plików uruchomić każdy algorytm 10 razy

7. Wyniki

Kolorem zielonym oznaczono najlepsze znalezione rozwiązanie, a kolorem czerwonym najgorsze. W nawiasach znajdujących się przy nazwie pliku podano najlepsze kiedykolwiek znalezione rozwiązania.

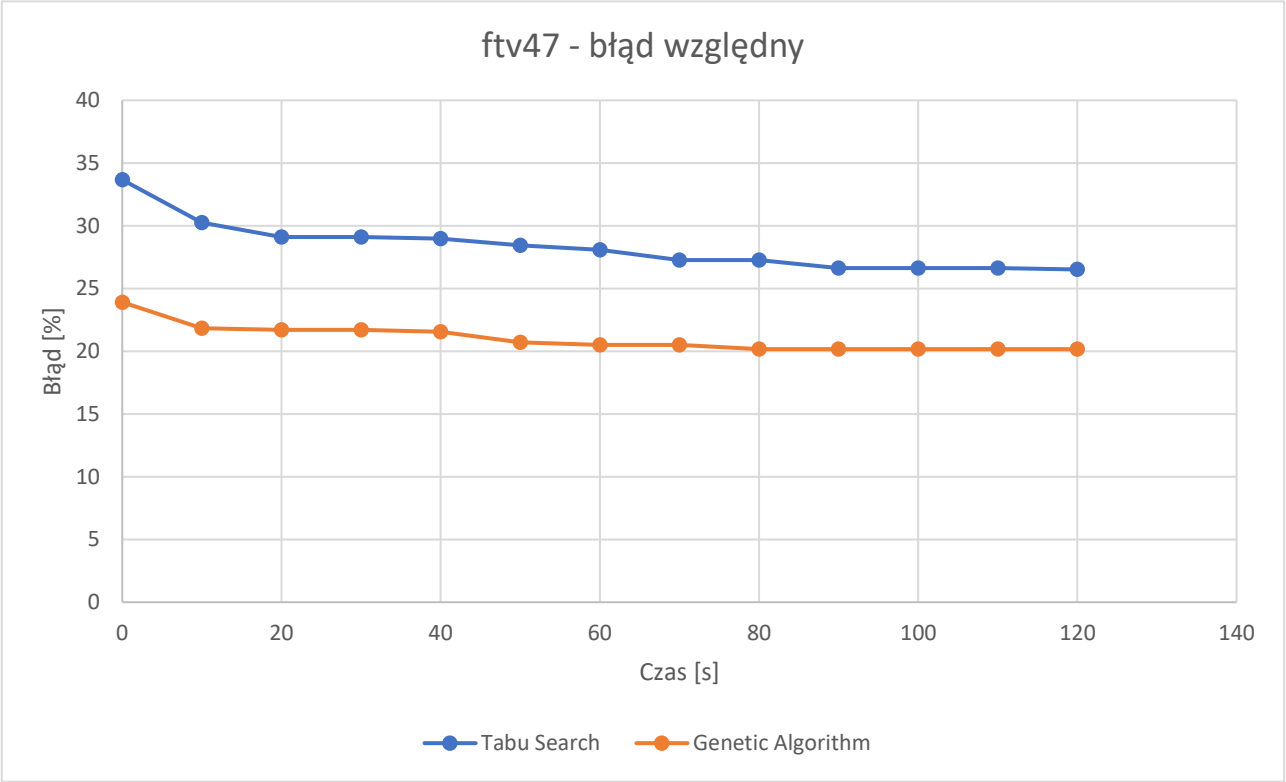
ftv47(1776) - 120 sekund							
Przeszukiwanie z zakazami				Algorytm genetyczny			
Wywołanie	Koszt	Czas znalezienia [s]	Błąd względny [%]	Wywołanie	Koszt	Czas znalezienia [s]	Błąd względny [%]
1	2265	59,49	27,53	1	2146	48,07	20,83
2	2320	0,07	30,63	2	2114	52,76	19,03
3	2231	0,85	25,62	3	2073	46,71	16,72
4	2256	2,71	27,03	4	2122	40,45	19,48
5	2248	15,69	26,58	5	2135	47,60	20,21
6	2299	32,68	29,45	6	2196	75,80	23,65
7	2207	41,46	24,27	7	2054	48,87	15,65
8	2205	15,08	24,16	8	2215	0,64	24,72
9	2200	61,17	23,87	9	2152	78,73	21,17
10	2183	117,10	22,92	10	2135	0,59	20,21

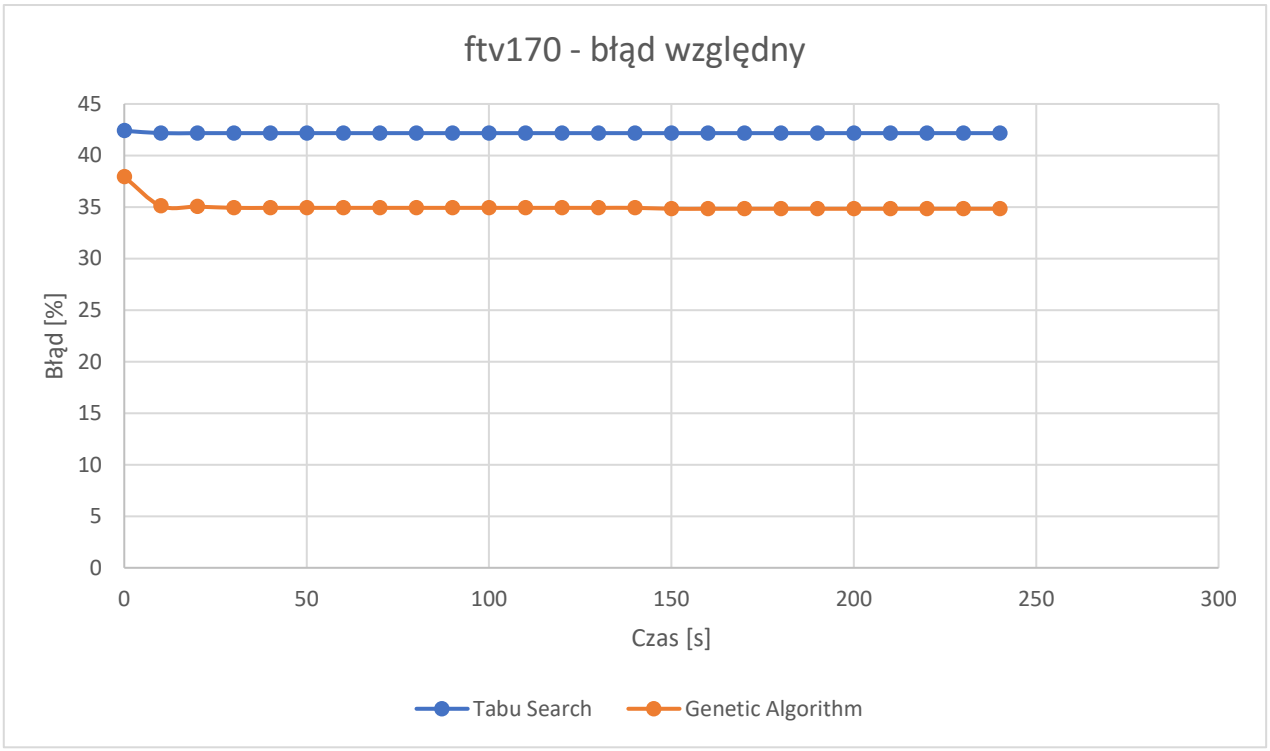
ftv170(2755) - 240 sekund							
Przeszukiwanie z zakazami				Algorytm genetyczny			
Wywołanie	Koszt	Czas znalezienia [s]	Błąd względny [%]	Wywołanie	Koszt	Czas znalezienia [s]	Błąd względny [%]
1	3888	0,14	41,13	1	3688	3,36	33,87
2	3923	0	42,4	2	3761	4,20	36,52
3	3923	0	42,4	3	3723	1,14	35,14
4	3923	0	42,4	4	3567	151,39	29,47
5	3923	0	42,4	5	3685	4,20	33,76
6	3895	0,28	41,38	6	3768	11,17	36,77
7	3923	0	42,4	7	3808	2,14	38,22
8	3923	0	42,4	8	3677	2,17	33,47
9	3923	0	42,4	9	3770	3,29	36,84
10	3923	0	42,4	10	3699	1,06	34,26

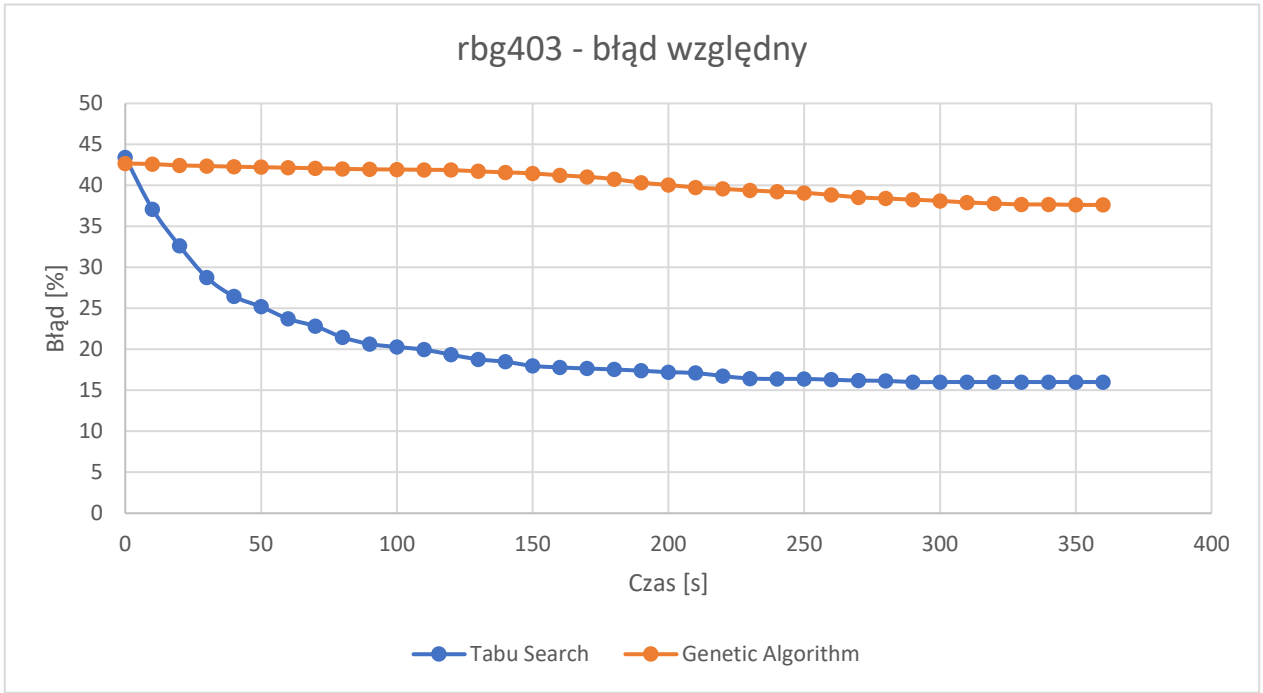
rbg403(2465) - 360 sekund							
Przeszukiwanie z zakazami				Algorytm genetyczny			
Wywołanie	Koszt	Czas znalezienia [s]	Błąd względny [%]	Wywołanie	Koszt	Czas znalezienia [s]	Błąd względny [%]
1	2873	143,77	16,55	1	3482	33,32	41,26
2	2847	223,80	15,5	2	3507	5,83	42,27
3	2863	134,47	16,15	3	3518	55,65	42,72
4	2867	225,87	16,31	4	3383	347,38	37,24
5	2874	271,34	16,59	5	3507	359,42	42,27
6	2808	232,35	13,91	6	3499	18,82	41,95
7	2846	175,18	15,46	7	3230	357,81	31,03
8	2880	96,48	16,84	8	3251	355,89	31,89
9	2858	286,68	15,94	9	3247	355,58	31,72
10	2881	144,31	16,88	10	3299	242,22	33,83

8. Wykresy

Przedstawione niżej wykresy przedstawiają zależność błędów względnych w zależności od czasu znalezienia rozwiązania




ftv170 - błąd względny
45
40
35
30
25
20
15
10
5
0
Błąd [%]
0
50
100
150
200
250
300
Czas [s]
Tabu Search
Genetic Algorithm


rbg403 - błąd względny
50
45
40
35
30
25
20
15
10
5
0
Błąd [%]
0
50
100
150
200
250
300
350
400
Czas [s]
Tabu Search
Genetic Algorithm

9. Podsumowanie

Na podstawie wykresów średnich serii błędu względnego od czasu można zauważyć, że algorytm genetyczny osiąga lepsze wyniki dla plików małego i średniego. Dla pliku dużego zdecydowanie lepiej wypada algorytm przeszukiwania z zakazami. Spowodowane jest to najprawdopodobniej różnicą czasu operacji jednej pętli algorytmu.

Algorytm przeszukiwania z zakazami podmienia po prostu dwa wierzchołki za to algorytm genetyczny tworzy wiele potomków co skutkuje znacznym wydłużeniem czasu pojedynczej pętli.

Wraz ze wzrostem ilości miast wydłuża się czas generacji kolejnej populacji co powoduje mniejszy zakres przeszukiwań. W celu ociągnięciu lepszych wyników należałoby przeznaczyć więcej czasu dla plików zawierających większa liczbę miast.

10. Bibliografia

https://pl.wikipedia.org/wiki/Algorytm_genetyczny

http://aragorn.pb.bialystok.pl/~wkwedlo/EA5.pdf

http://www.zio.iiar.pwr.wroc.pl/pea/w9_ga_tsp.pdf