Sprawozdanie:

Projektowanie Efektywnych Algorytmów

Dmytro Verkhovsky 231844

**Prowadzący:** dr inż. Dariusz Banasiak

# Zadanie projektowe numer 3

Algorytm Genetyczny dla problemu komiwojażera

08.01.2018

## **Wstęp**

* 1. **Opis zadania projektowego**

Należało zaimplementować Genetyczny algorytm dla problemu komiwojażera oraz wykonać pomiary czasu działania algorytmu w zależności od wielkości instancji oraz jakości dostarczanych rozwiązań a po tym rozwiązania porównywać z najlepszymi znanymi rozwiązaniami dla konkretnych danych testowych.

**1.2 Wstęp teoretyczny**

1.2.1 *Problem komiwojażera*

Problem komiwojażera jest zagadnieniem optymalizacyjnym, polegającym na znalezieniu minimalnego cyklu Hamiltona w pełnym graﬁe ważonym. Nazwa pochodzi od typowej ilustracji problemu, przedstawiającej go z punktu widzenia wędrownego sprzedawcy (komiwojażera): dane jest n miast, które komiwojażer ma odwiedzić, oraz odległość / cena podróży / czas podróży pomiędzy każdą parą miast. Celem jest znalezienie najkrótszej / najtańszej / najszybszej drogi łączącej wszystkie miasta, zaczynającej się i kończącej się w określonym punkcie.

1.2.2 *Genetyczny Algorytm*

Heurystyka metoda znajdowania rozwiązań, dla której nie ma gwarancji znalezienia rozwiązania [optymalnego](https://pl.wikipedia.org/wiki/Optymalizacja), a często nawet prawidłowego.

Algorytm genetyczny – rodzaj [heurystyki](https://pl.wikipedia.org/wiki/Heurystyka_(informatyka)" \o "Heurystyka (informatyka)) przeszukującej przestrzeń alternatywnych rozwiązań problemu w celu wyszukania rozwiązań najlepszych. Sposób działania algorytmów genetycznych nieprzypadkowo przypomina zjawisko [ewolucji biologicznej](https://pl.wikipedia.org/wiki/Ewolucja" \o "Ewolucja), ponieważ ich twórca [John Henry Holland](https://pl.wikipedia.org/wiki/John_Henry_Holland" \o "John Henry Holland) właśnie z biologii czerpał inspiracje do swoich prac. Obecnie zalicza się go do grupy algorytmów ewolucyjnych.

Rozwój Algorytmu Genetycznego

**1957-62**: Barricelli, Fraser, Martin, Cockerham — modelowanie procesów genetycznych

**1962:** Holland — proponowanie formalizmu systemu adaptacyjnego i strategii adaptacyjnych zwanych "adaptive plans"(„planami adapcyjnymi”)

**1967:** Bagley  — badał meta-środowiska i genetyczny plan adaptacyjny, nazywany algorytmem genetycznym stosowanym do prostej gry o nazwie hexapawn

**1971:** Hollstien — badał plany genetyczne dotyczące kontroli adaptacyjnej i optymalizacji funkcji

**1975:** De Jong przeprowadził gruntowne badanie genetycznego modelu adaptacyjnego (genetic plans) stosowanego do optymalizacji funkcji ciągłych, a jego zestaw problemów testowych jest nadal powszechnie stosowany

**1985:** Goldberg — optymalizacja pracy gazociągu

1.2.3 Przykłady i zastosowania:

Algorytmy genetyczne znalazły zastosowanie do rozwišzywania problemów optymalizacji, rozpoznawania obrazów, przewidywaniu ruchów na giełdzie, tworzenia grafiki oraz projektowania budynków i maszyn.

*Ordered Crossover – OX(Davis,1985)*

OX buduje potomstwo, wybierając podsekcję trasy od jednego rodzica, zachowując względną kolejność pozycji od drugiego rodzica. Począwszy od drugiego punktu jednego rodzica, pozycje drugiego rodzica są kopiowane w tej samej kolejności, pomijając symbole, które już są obecne. Osiągając koniec łańcucha, kontynuujemy od początku łańcucha.

Przykładowe rozwiązanie danych przy użyczyću OX:

P1: 142**|**8573**|**69

P2: 753**|**1986**|**42

O1: xxx**|**8573**|**xx

O2: xxx**|**1986**|**xx

O1:196**|**8573**|**42

O2:573**|**1986**|**42

W powyższym przykładzie wypełniono chromosom O2 zgodnie z sekwencją

6 -> 9 -> 1 -> 4 -> 2 -> 8 -> 5 -> 7 -> 3 i wypełniając pozycje jeszcze nieobecne.

Robimy to samo dla chromosomu O1 na liście 4 -> 2 -> 7 -> 5 ->3 -> 1 -> 9 -> 8 -> 6

i wypełniamy pozycji jeszcze nieobecne.

*Selekcja :*

Metoda rangowa:

Stopień przystosowania jest reprezentowany przez rangę, która nadawana jest w procesie sortowania osobników w populacji.

Sortowanie polega na uszeregowaniu osobników w populacji według nierosnących wartości funkcji przystosowania.

Dwa sposoby przypisywania rang: – ranga jest równa indeksowi (pozycji) w uszeregowanej populacji – ranga identyczna dla osobników identycznych.

*Mutacja:*

inversion - wybiera losowo podciągi miast i zamienia ich kolejność.

p = ( 1 2 ∣ 3 4 5 6 ∣ 7 8 9 ) -> p ′ = ( 1 2 ∣ 6 5 4 3 ∣ 7 8 9 )

**1.3 Rozpatrywanie algorytmu**

1. Wyznaczenie populacji ***V*** (zbioru losowych rozwiązań)
2. Pętla – dopóki nie wystąpi **zdarzenie krytyczne *X* (**zadana ***liczba pokoleń*)**:
   1. Dokonaj selekcji metodą rankingową
   2. Wykonaj krzyżowanie metodą *OX* (z prawdopodobieństwem ***Y***)
   3. Zmutuj nowe osobniki (z prawdopodobieństwem ***Z***), biorąc podciąg miast i odwracając jego kolejność (*invert*)
   4. Dodaj nowe rozwiązania do populacji
   5. Posortuj populację wg rozwiązań i przytnij do zadanego rozmiaru (***V***)

Zakończ. Pierwsze rozwiązanie z populacji (najmniejsze) jest naszym rozwiązaniem.

**2. Plan eksperymentu**

- Czasy poszczególnych operacji mierzone są przy pomocy funkcji zawartych w bibliotece **<chrono>** będącej częścią standardu C++11.

- Będziemy też wczytywać już przygotowane dane z strony zamieszczonej w pliku.

-Test jaki mierzy czas dla rożnej ilości miast jest przeprowadzony 100 dla konkretnego rozmiaru miasta.

**3. Wyniki eksperymentu**

Wyniki czasowe dla rożnych ilości miast:

Br17(atsp)

Najlepszy wynik w TSPLIB: 39

Najlepszy wynik uzyskany za pomocą programowania dynamicznego: 40 (2,91%)

Najlepszy wynik uzyskany za pomocą tabu search: 40 (2,91%)

Najlepszy wynik uzyskany za pomocą algorytmu genetycznego: 39 (0,00%)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Czas(s) | Populacja | Licz.pokoleń | Praw.Krzyżownia | Praw.Mutacji | Odleglosc | Bląd |
| 0,009844 | 40 | 100 | 0.7 | 0.3 | 40 | 2,56% |
| 0,009632 | 40 | 100 | 0.8 | 0.2 | 39 | 0% |
| 0,011799 | 40 | 100 | 0.9 | 0.1 | 39 | 0% |
| 0,019250 | 40 | 200 | 0.7 | 0.3 | 40 | 2,56% |
| 0,020103 | 40 | 200 | 0.8 | 0.2 | 39 | 0% |
| 0,021458 | 40 | 200 | 0.9 | 0.1 | 39 | 0% |
| 0,03866 | 40 | 400 | 0.7 | 0.3 | 39 | 0% |
| 0,039703 | 40 | 400 | 0.8 | 0.2 | 39 | 0% |
| 0,040129 | 40 | 400 | 0.9 | 0.1 | 39 | 0% |
| 0,0711043 | 40 | 800 | 0.7 | 0.3 | 39 | 0% |
| 0,0777493 | 40 | 800 | 0.8 | 0.2 | 39 | 0% |
| 0,085802 | 40 | 800 | 0.9 | 0.1 | 39 | 0% |
| 0,132813 | 40 | 1600 | 0.7 | 0.3 | 39 | 0% |
| 0,146481 | 40 | 1600 | 0.8 | 0.2 | 39 | 0% |
| 0,1560570 | 40 | 1600 | 0.9 | 0.1 | 39 | 0% |
| 0,0202737 | 80 | 100 | 0.7 | 0.3 | 39 | 0% |
| 0,0221434 | 80 | 100 | 0.8 | 0.2 | 39 | 0% |
| 0,024507 | 80 | 100 | 0.9 | 0.1 | 39 | 0% |
| 0,0393611 | 80 | 200 | 0.7 | 0.3 | 39 | 0% |
| 0,042114 | 80 | 200 | 0.8 | 0.2 | 39 | 0% |
| 0,043203 | 80 | 200 | 0.9 | 0.1 | 39 | 0% |
| 0,0697723 | 80 | 400 | 0.7 | 0.3 | 39 | 0% |
| 0,074390 | 80 | 400 | 0.8 | 0.2 | 39 | 0% |
| 0,080927 | 80 | 400 | 0.9 | 0.1 | 39 | 0% |
| 0,131841 | 80 | 800 | 0.7 | 0.3 | 39 | 0% |
| 0,149339 | 80 | 800 | 0.8 | 0.2 | 39 | 0% |
| 0,165061 | 80 | 800 | 0.9 | 0.1 | 39 | 0% |
| 0,266961 | 80 | 1600 | 0.7 | 0.3 | 39 | 0% |
| 0,2983751 | 80 | 1600 | 0.8 | 0.2 | 39 | 0% |
| 0,325899 | 80 | 1600 | 0.9 | 0.1 | 39 | 0% |

gr17(tsp)

Najlepszy wynik w TSPLIB: 2085

Najlepszy wynik uzyskany za pomocą symulowanego wyżarzania: 2150 (3,12%)

Najlepszy wynik uzyskany za pomocą tabu search: 2112 (1,31%)

Najlepszy wynik uzyskany za pomocą algorytmu genetycznego: 2085 (0,00%)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Czas(s) | Populacja | Licz.pokoleń | Praw.Krzyżownia | Praw.Mutacji | Odleglosc | Bląd |
| 0,009342 | 40 | 100 | 0.7 | 0.3 | 2224 | 6,71% |
| 0,00989 | 40 | 100 | 0.8 | 0.2 | 2205 | 5,79% |
| 0,01199 | 40 | 100 | 0.9 | 0.1 | 2188 | 4,94% |
| 0,01769 | 40 | 200 | 0.7 | 0.3 | 2203 | 5,75% |
| 0,02009 | 40 | 200 | 0.8 | 0.2 | 2185 | 4,92% |
| 0,02198 | 40 | 200 | 0.9 | 0.1 | 2172 | 4,14% |
| 0,03751 | 40 | 400 | 0.7 | 0.3 | 2198 | 5,41% |
| 0,03905 | 40 | 400 | 0.8 | 0.2 | 2178 | 4,46% |
| 0,043213 | 40 | 400 | 0.9 | 0.1 | 2169 | 3,95% |
| 0,07321 | 40 | 800 | 0.7 | 0.3 | 2175 | 4,31% |
| 0,07798 | 40 | 800 | 0.8 | 0.2 | 2163 | 3,78% |
| 0,08503 | 40 | 800 | 0.9 | 0.1 | 2146 | 2,93% |
| 0,12953 | 40 | 1600 | 0.7 | 0.3 | 2160 | 3,74% |
| 0,14217 | 40 | 1600 | 0.8 | 0.2 | 2156 | 3,35% |
| 0,16070 | 40 | 1600 | 0.9 | 0.1 | 2153 | 3,30% |
| 0,01814 | 80 | 100 | 0.7 | 0.3 | 2148 | 3,06% |
| 0,02134 | 80 | 100 | 0.8 | 0.2 | 2137 | 2,44% |
| 0,02397 | 80 | 100 | 0.9 | 0.1 | 2132 | 2,25% |
| 0,0380 | 80 | 200 | 0.7 | 0.3 | 2130 | 2,20% |
| 0,0414 | 80 | 200 | 0.8 | 0.2 | 2118 | 1,58% |
| 0,0571 | 80 | 200 | 0.9 | 0.1 | 2116 | 1,47% |
| 0,0764 | 80 | 400 | 0.7 | 0.3 | 2116 | 1,47% |
| 0,0817 | 80 | 400 | 0.8 | 0.2 | 2102 | 0,85% |
| 0,0894 | 80 | 400 | 0.9 | 0.1 | 2103 | 0,86% |
| 0,1271 | 80 | 800 | 0.7 | 0.3 | 2103 | 0,86% |
| 0,1399 | 80 | 800 | 0.8 | 0.2 | 2085 | 0% |
| 0,1736 | 80 | 800 | 0.9 | 0.1 | 2085 | 0% |
| 0,2661 | 80 | 1600 | 0.7 | 0.3 | 2085 | 0% |
| 0,2871 | 80 | 1600 | 0.8 | 0.2 | 2085 | 0% |
| 0,3158 | 80 | 1600 | 0.9 | 0.1 | 2085 | 0% |

ft53(atsp)

Najlepszy wynik w TSPLIB: 6905

Najlepszy wynik uzyskany za pomocą tabu search: 8395 (21,59%)

Najlepszy wynik uzyskany za pomocą algorytmu genetycznego: 7246 (4,938%)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Czas(s) | Populacja | Licz.pokoleń | Praw.Krzyżownia | Praw.Mutacji | Odleglosc | Bląd |
| 0.229 | 80 | 800 | 0,7 | 0,3 | 9456 | 36,9442% |
| 0.2405 | 80 | 800 | 0,8 | 0,2 | 9365 | 35,6264% |
| 0.2788 | 80 | 800 | 0,9 | 0,1 | 9318 | 34,9457% |
| 0.4955 | 80 | 1600 | 0,7 | 0,3 | 9144 | 32,4258% |
| 0,5083 | 80 | 1600 | 0,8 | 0,2 | 8915 | 29,1093% |
| 0,5484 | 80 | 1600 | 0,9 | 0,1 | 8877 | 28,559% |
| 1,0062 | 80 | 3200 | 0,7 | 0,3 | 8770 | 27,009% |
| 1,2166 | 80 | 3200 | 0,8 | 0,2 | 8671 | 25,575% |
| 1,3669 | 80 | 3200 | 0,9 | 0,1 | 8586 | 24,344% |
| 0,5317 | 160 | 800 | 0,7 | 0,3 | 9213 | 33,425% |
| 0,5954 | 160 | 800 | 0,8 | 0,2 | 9046 | 31,0065% |
| 0,6700 | 160 | 800 | 0,9 | 0,1 | 9022 | 30,6589% |
| 0,971 | 160 | 1600 | 0,7 | 0,3 | 8835 | 27,95% |
| 1,108 | 160 | 1600 | 0,8 | 0,2 | 8642 | 25,155% |
| 1,614 | 160 | 1600 | 0,9 | 0,1 | 8532 | 23,562% |
| 2,082 | 160 | 3200 | 0,7 | 0,3 | 8478 | 22,780% |
| 2,123 | 160 | 3200 | 0,8 | 0,2 | 8283 | 19,956% |
| 2,394 | 160 | 3200 | 0,9 | 0,1 | 8216 | 18,986% |
| 1,035 | 320 | 800 | 0,7 | 0,3 | 8086 | 17.103% |
| 1,155 | 320 | 800 | 0,8 | 0,2 | 8030 | 16,292% |
| 1,392 | 320 | 800 | 0,9 | 0,1 | 7883 | 14,163% |
| 2,039 | 320 | 1600 | 0,7 | 0,3 | 7854 | 13,743% |
| 2,367 | 320 | 1600 | 0,8 | 0,2 | 7730 | 11,947% |
| 2,748 | 320 | 1600 | 0,9 | 0,1 | 7581 | 9,790% |
| 4,931 | 320 | 3200 | 0,7 | 0,3 | 7427 | 7,559% |
| 5,319 | 320 | 3200 | 0,8 | 0,2 | 7338 | 6,270% |
| 6,213 | 320 | 3200 | 0,9 | 0,1 | 7246 | 4,938% |

ftv55(atsp)

Najlepszy wynik w TSPLIB: 1608

Najlepszy wynik uzyskany za pomocą tabu search: 2004 (24,66%)

Najlepszy wynik uzyskany za pomocą algorytmu genetycznego: 2134 (32,71%)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Czas(s) | Populacja | Licz.pokoleń | Praw.Krzyżownia | Praw.Mutacji | Odleglosc | Bląd |
| 0,2192 | 80 | 800 | 0,7 | 0,3 | 2397 | 49,8134% |
| 0,2351 | 80 | 800 | 0,8 | 0,2 | 2394 | 48,8806% |
| 0,2732 | 80 | 800 | 0,9 | 0,1 | 2338 | 45,398% |
| 0,4724 | 80 | 1600 | 0,7 | 0,3 | 2301 | 43,097% |
| 0,5012 | 80 | 1600 | 0,8 | 0,2 | 2284 | 42,0398% |
| 0,5692 | 80 | 1600 | 0,9 | 0,1 | 2251 | 39,9876% |
| 1,2532 | 80 | 3200 | 0,7 | 0,3 | 2243 | 39,49% |
| 1,3121 | 80 | 3200 | 0,8 | 0,2 | 2190 | 36,194% |
| 1,4952 | 80 | 3200 | 0,9 | 0,1 | 2157 | 34,1418% |
| 0.6021 | 160 | 800 | 0,7 | 0,3 | 2139 | 33,0224% |
| 0,6341 | 160 | 800 | 0,8 | 0,2 | 2115 | 31.5299% |
| 0,697 | 160 | 800 | 0,9 | 0,1 | 2090 | 29.9751% |
| 1.099 | 160 | 1600 | 0,7 | 0,3 | 2089 | 29,9129% |
| 1,379 | 160 | 1600 | 0,8 | 0,2 | 2071 | 28,7935% |
| 1,474 | 160 | 1600 | 0,9 | 0,1 | 2043 | 27,0522% |
| 2,233 | 160 | 3200 | 0,7 | 0,3 | 2038 | 26,7413% |
| 2,6536 | 160 | 3200 | 0,8 | 0,2 | 2002 | 24,5025% |
| 2,7049 | 160 | 3200 | 0,9 | 0,1 | 1963 | 22,0771% |
| 1,0688 | 320 | 800 | 0,7 | 0,3 | 1943 | 20,833% |
| 1,30828 | 320 | 800 | 0,8 | 0,2 | 1917 | 19,2164% |
| 1,3743 | 320 | 800 | 0,9 | 0,1 | 1909 | 18,718% |
| 2.397 | 320 | 1600 | 0,7 | 0,3 | 1900 | 18.17% |
| 2,566 | 320 | 1600 | 0,8 | 0,2 | 1882 | 17,0398% |
| 2,789 | 320 | 1600 | 0,9 | 0,1 | 1869 | 16,2313% |
| 4,213 | 320 | 3200 | 0,7 | 0,3 | 1850 | 15,0498% |
| 4,3164 | 320 | 3200 | 0,8 | 0,2 | 1824 | 13,4328% |
| 4,3929 | 320 | 3200 | 0,9 | 0,1 | 1797 | 11,7537% |

a280(tsp)

Najlepszy wynik w TSPLIB: 2579

Najlepszy wynik uzyskany za pomocą tabu search: 3116 (20,86%)

Najlepszy wynik uzyskany za pomocą algorytmu genetycznego: 9868 (279,80%)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Czas(s) | Populacja | Licz.pokoleń | Praw.Krzyżownia | Praw.Mutacji | Odleglosc | Bląd |
| 1,2872 | 80 | 800 | 0,7 | 0,3 | 9312 | 261,07% |
| 1.4578 | 80 | 800 | 0,8 | 0,2 | 9087 | 252,34% |
| 1,7677 | 80 | 800 | 0,9 | 0,1 | 8956 | 247,26% |
| 2,8951 | 80 | 1600 | 0,7 | 0,3 | 8869 | 243,89% |
| 3,416 | 80 | 1600 | 0,8 | 0,2 | 8755 | 239,47% |
| 4,992 | 80 | 1600 | 0,9 | 0,1 | 8484 | 228,96% |
| 5,514 | 80 | 3200 | 0,7 | 0,3 | 8426 | 226,71% |
| 7,791 | 80 | 3200 | 0,8 | 0,2 | 8352 | 223,84% |
| 8,3952 | 80 | 3200 | 0,9 | 0,1 | 8258 | 220,22% |
| 2,7905 | 160 | 800 | 0,7 | 0,3 | 8252 | 219,96% |
| 3,0639 | 160 | 800 | 0,8 | 0,2 | 7988 | 209,73% |
| 3,4806 | 160 | 800 | 0,9 | 0,1 | 7817 | 203,102% |
| 6,33 | 160 | 1600 | 0,7 | 0,3 | 5907 | 129,04% |
| 7,17 | 160 | 1600 | 0,8 | 0,2 | 5775 | 123,92% |
| 7,989 | 160 | 1600 | 0,9 | 0,1 | 5720 | 121,79% |
| 10,85 | 160 | 3200 | 0,7 | 0,3 | 5845 | 126,63% |
| 12,398 | 160 | 3200 | 0,8 | 0,2 | 5718 | 121,71% |
| 15,14 | 160 | 3200 | 0,9 | 0,1 | 5595 | 116,94% |
| 7,91 | 320 | 800 | 0,7 | 0,3 | 7958 | 208,56% |
| 7,766 | 320 | 800 | 0,8 | 0,2 | 7835 | 203,8% |
| 7,69 | 320 | 800 | 0,9 | 0,1 | 7349 | 184,95% |
| 14,02 | 320 | 1600 | 0,7 | 0,3 | 5322 | 106,35% |
| 15,67 | 320 | 1600 | 0,8 | 0,2 | 5256 | 103,8% |
| 16,26 | 320 | 1600 | 0,9 | 0,1 | 5111 | 98,17% |
| 33,14 | 320 | 3200 | 0,7 | 0,3 | 4672 | 81,15% |
| 28,42 | 320 | 3200 | 0,8 | 0,2 | 4641 | 79,95% |
| 24,94 | 320 | 3200 | 0,9 | 0,1 | 4328 | 67,81% |

Pr439(tsp)

Najlepszy wynik w TSPLIB: 107217

Najlepszy wynik uzyskany za pomocą tabu search: 130150 (21,39%)

Najlepszy wynik uzyskany za pomocą algorytmu genetycznego: 234883 (129,57%)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Czas(s) | Populacja | Licz.pokoleń | Praw.Krzyżownia | Praw.Mutacji | Odleglosc | Bląd |
| 2,3114 | 80 | 800 | 0,7 | 0,3 | 611823 | 498,53% |
| 3,5743 | 80 | 800 | 0,8 | 0,2 | 594050 | 481,16% |
| 4,2643 | 80 | 800 | 0,9 | 0,1 | 581235 | 468,62% |
| 6,02219 | 80 | 1600 | 0,7 | 0,3 | 444896 | 335,24% |
| 7,20159 | 80 | 1600 | 0,8 | 0,2 | 431936 | 322,56% |
| 8,1647 | 80 | 1600 | 0,9 | 0,1 | 429282 | 319,97% |
| 10,1893 | 80 | 3200 | 0,7 | 0,3 | 1163304 | 299,38% |
| 12.977 | 80 | 3200 | 0,8 | 0,2 | 397316 | 288,69% |
| 14,597 | 80 | 3200 | 0,9 | 0,1 | 383332 | 275,01% |
| 5,91 | 160 | 800 | 0,7 | 0,3 | 597339 | 484,38% |
| 6,53 | 160 | 800 | 0,8 | 0,2 | 573919 | 461,47% |
| 7,87 | 160 | 800 | 0,9 | 0,1 | 563844 | 451,15% |
| 11,49 | 160 | 1600 | 0,7 | 0,3 | 382093 | 273,80% |
| 13,72 | 160 | 1600 | 0,8 | 0,2 | 369535 | 261,52% |
| 15,099 | 160 | 1600 | 0,9 | 0,1 | 353945 | 246,26% |
| 23,76 | 160 | 3200 | 0,7 | 0,3 | 310955 | 204,11% |
| 24,73 | 160 | 3200 | 0,8 | 0,2 | 301987 | 195,37% |
| 26,48 | 160 | 3200 | 0,9 | 0,1 | 296619 | 190,86% |
| 28,45 | 320 | 800 | 0,7 | 0,3 | 575744 | 463,25% |
| 30,03 | 320 | 800 | 0,8 | 0,2 | 563777 | 451,49% |
| 31,19 | 320 | 800 | 0,9 | 0,1 | 556094 | 444,03% |
| 36,99 | 320 | 1600 | 0,7 | 0,3 | 332607 | 225,39% |
| 38,34 | 320 | 1600 | 0,8 | 0,2 | 321793 | 214,81% |
| 40,19 | 320 | 1600 | 0,9 | 0,1 | 308352 | 201,66% |
| 43,29 | 320 | 3200 | 0,7 | 0,3 | 261190 | 155,53% |
| 45,19 | 320 | 3200 | 0,8 | 0,2 | 242799 | 137,53% |
| 47,34 | 320 | 3200 | 0,9 | 0,1 | 234883 | 129,57% |

**Czas dla wybranych plików gdzie populacja jest równa 80, liczba pokoleń jest równa 800, a prawdopodobieństwo krzyżowania i mutacji są podane na wykresie.**

**Błąd dla wybranych plików gdzie populacja jest równa 80, liczba pokoleń jest równa 800, a prawdopodobieństwo krzyżowania i mutacji są podane na wykresie.**

**Czas dla wybranych plików gdzie populacja jest równa 80, liczba pokoleń jest równa 3200, a prawdopodobieństwo krzyżowania i mutacji są podane na wykresie.**

**Czas dla wybranych plików gdzie populacja jest równa 320, liczba pokoleń jest równa 3200, a prawdopodobieństwo krzyżowania i mutacji są podane na wykresie.**

**Błąd dla wybranych plików gdzie populacja jest równa 80, liczba pokoleń jest równa 3200, a prawdopodobieństwo krzyżowania i mutacji są podane na wykresie.**

**Błąd dla wybranych plików gdzie populacja jest równa 320, liczba pokoleń jest równa 3200, a prawdopodobieństwo krzyżowania i mutacji są podane na wykresie.**

Na powyższym wykresie możemy zobaczyć różnicę między czasem działania Algorytmu programowania dynamicznego, Tabu Search i Algorytmu Genetycznego.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Programowanie dynamiczne | | Tabu Search | | Algorytm Genetyczny | |
| Ilość miast | Czas [s] | Czas [s] | Error % | Czas [s] | Error % |
| 17(br17.atsp) | 0,110605 | 0,005 | 2,915 | 0,0099 | 0 |
| 17(gr17.tsp) | 0,208331 | 0,0066 | 1,311 | 0,0093 | 6,71 |

**4. Wnioski**

W tym projekcie zostały wykonane pomiary dla algorytmu Genetycznego dla problemu Komiwojazera .

Z powyższego wykresu możemy zobaczyć że algorytm Genetycznego ma swoje wady i zalety.

Z jednej strony on jest szybki, ale z innej strony on ma w sobie ukryty błąd , jaki możemy zobaczyć w powyższej tabeli i ten błąd jest mały dla takich plików(gr17.tsp ma poprawną ściezkę o długość 2085, błąd dla prawie wszystkich był mały, mniejszy od 7%), ale się zwiększa dla dużych plików. Jeżeli dobrze ustawimy parametry za pomocą których będzie działać algorytm: rozmiar populacji, liczba pokoleń, prawdopodobieństwo krzyżowania i mutacji , to możemy uzyskać przybliżony wynik do poprawnego, ale im lepszy jest wynik im dłuższej wykonuje się algorytm. Jeszcze jedną zaletą jest to że tę algorytm, jak i algorytm Tabu Search, działa dla dużych plików(np. pr439) dla których programowanie dynamiczne już nie działa bo potrzebuje dużą ilość pamięći i nawet jeżeliby byłaby ta ilość pamięći to by trwało długo. Płacimy za tę szybki czas i małą ilość potrzebowanej pamięći dokładnością otrzymanych wyników.