

Lista 3

Tomasz Hałas i Szymon Szymecki

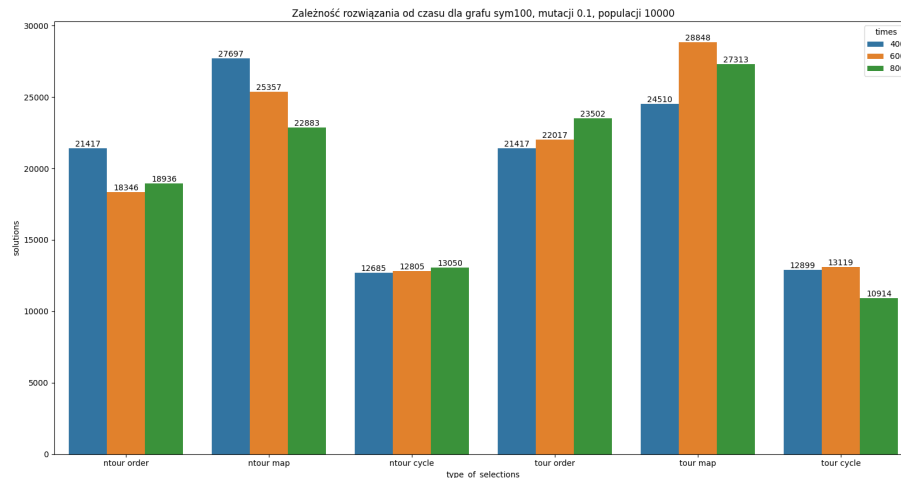
June 2022

1 Wstęp

Naszym celem jest zaimplementowanie algorytmów populacyjnych opartych na mechanizmie rozwiązywania problemu inspirowany biologicznym procesem doboru naturalnego i ewolucji.

2 Start

Z początku nasz algorytm wybiera populacje o rozmiarze k , gdzie k jest uzależnione od rozmiaru grafu, jako genotyp wstawiamy tablice o unikatowym losowych indeksach miasta z problemu TSP. Zrobiliśmy też test sprawdzający czy unikatowość losowania ma wpływ na zbieżność algorytmów.



Rysunek 1: Porównanie metody wyboru populacji początkowej

3 Selekcja

Selekcja (wybór) - bieżącej populacji rodziców, która będzie używana w procesie krzyżowania. Relizujemy ją w następujący sposób:

1. Losowa (jednorodna) – każdy osobnik ma równe szansa bycia wybranym
2. Ruletka – lepiej przystosowane osobniki mają większą szanse bycia wybranym niż słabsze
3. Turniej – wybór najlepszego osobnika z pośród 5 losowych osobników. Z 5 losowych osobników wybieramy najlepszego pod względem "distansu" w problemie TSP.

4 Krzyżowanie

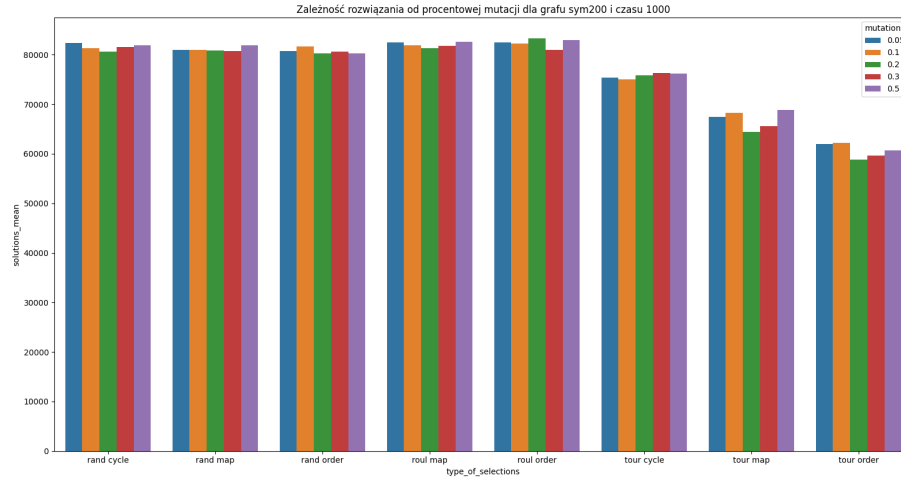
Krzyżowanie jest procesem tworzenia osobników potomnych na bazie osobników rodzicielskich. U nas zaimplementowane mamy 3 rodzaje krzyżowań:

1. Z zachowaniem porządku (Order Crossover, OX) - środkową częścią dziecka stanowi jeden rodzic, natomiast początek i koniec bierzemy od drugiego rodzica z zachowaniem kolejności występowania elementów w permutacji,
2. Z częściowym mapowaniem (Partially Mapped Crossover, PMX) - środkową częścią dziecka stanowi jeden rodzic, tworzymy mapowanie występujących elementów środkowych obydwu rodziców, następnie początek i koniec bierzemy od drugiego rodzica, zastępując użyte w środku elementy za pomocą wcześniejszego mapowania,
3. Cykliczny (Cycle Crossover, CX) - dziecko jest generowane poprzez dodawanie całych cykli występujących naprzemiennie u jednego i drugiego rodzica.

Każde zaimplementowane krzyżowanie po wybraniu dwóch rodziców generuje dwójkę dzieci, przy czym po wygenerowaniu pierwszego dziecka, rodzice są zamieniani ze sobą miejscami przy generowaniu drugiego dziecka.

5 Mutacja

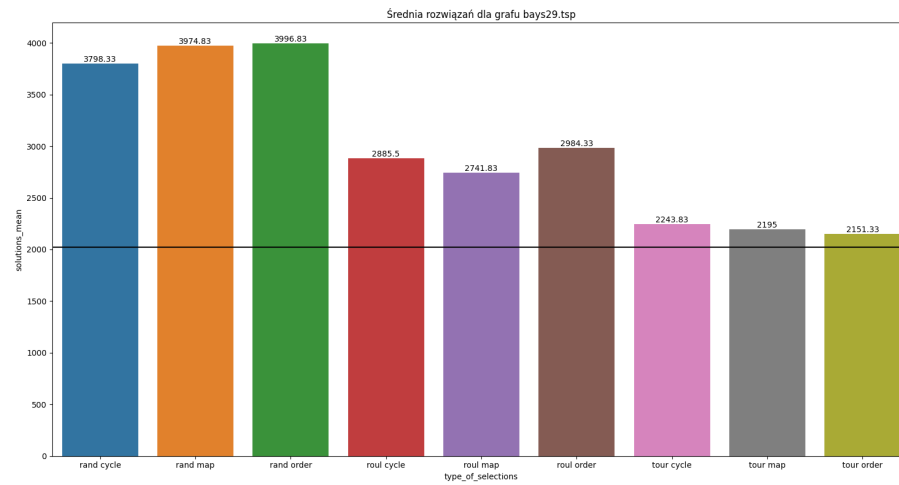
Z pewnym prawdopodobieństwem każdy osobnik może zostać poddany mutacji. U nas mutacja przebiega na każdego osobnika w indywidualny sposób dla każdego osobnika za pomocą inwersji. Prawdopodobieństwo zaisntienia inwersji jest losowe w odnosieniu do parametru k .



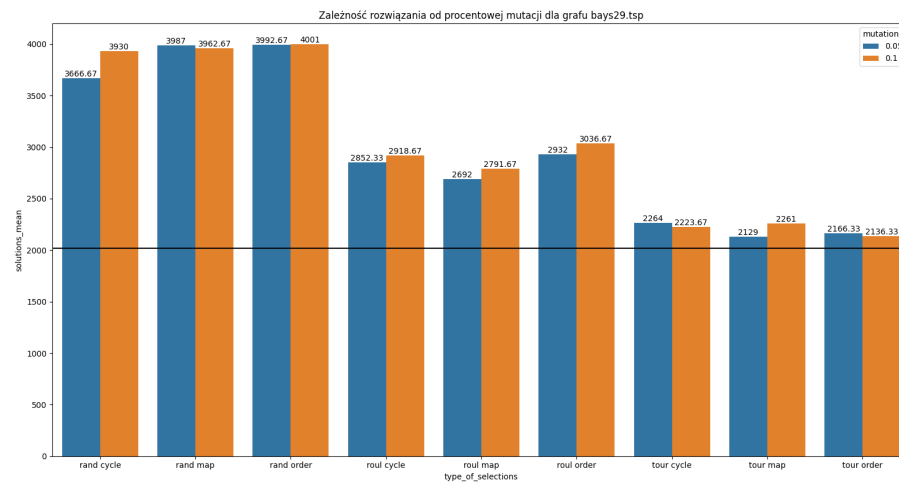
Rysunek 2: Wpływ mutacji na wyniki

Widzimy, że zbyt mała mutacja powoduje że rozwiązania są mniej zbieżne, wolniej osiągają rozwiązania (poprzez mutacje możemy wpaść szybciej z minimum lokalnego). Natomiast zbyt duża mutacja sprawia że oddalamy się od rozwiązania optymalnego (rozbieżność). Widzimy to na słupkach które przyjmują charakterystyczny łuk.

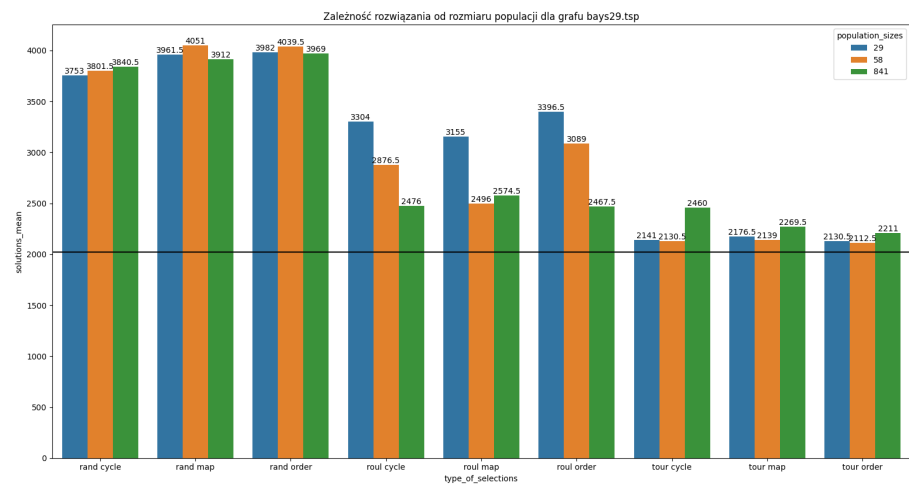
6 Poprawność implementacji



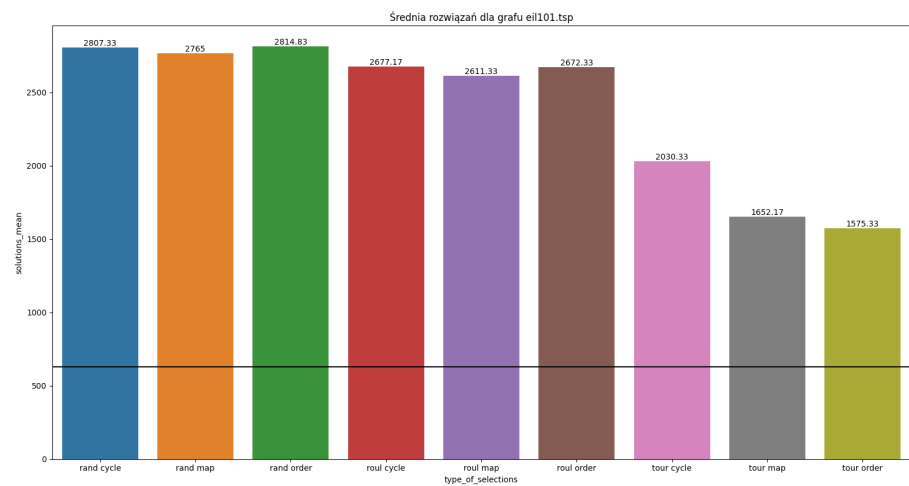
Rysunek 3: Średnie rozwiązania



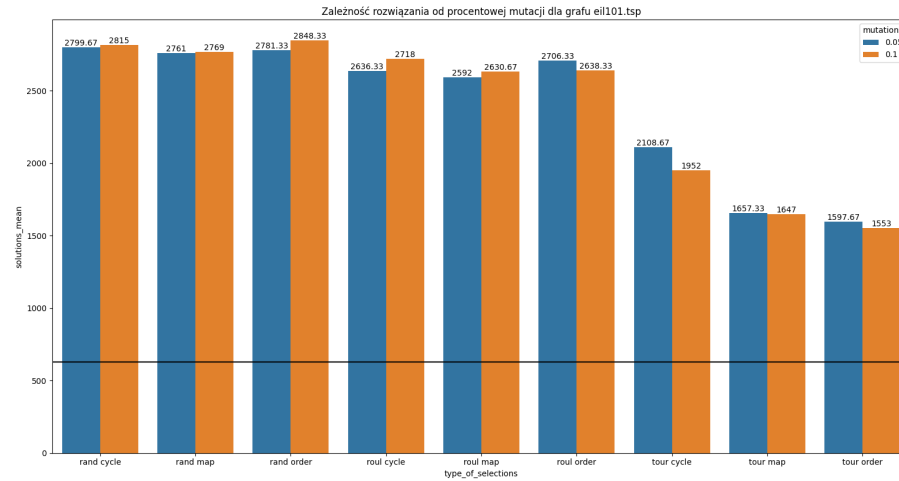
Rysunek 4: Działanie mutacji na problem bays29



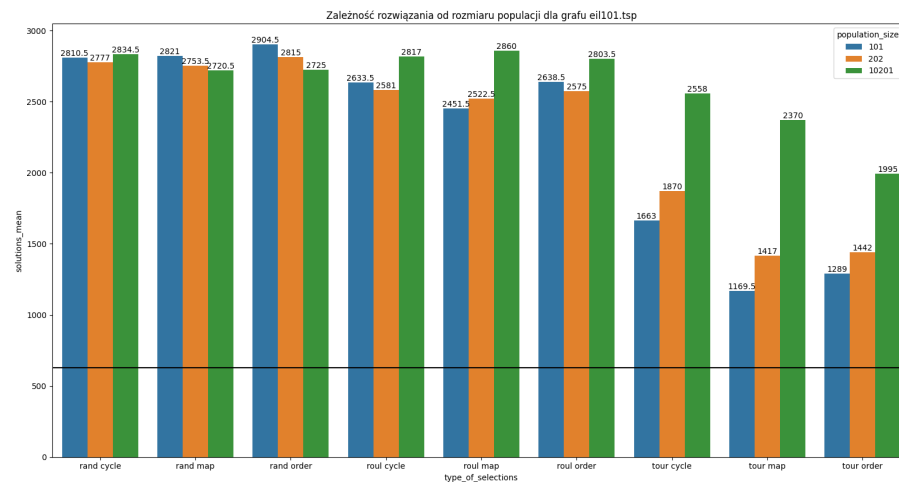
Rysunek 5: Działanie rozmiaru populacji na problem bays29



Rysunek 6: Średnie rozwiązania



Rysunek 7: Działanie mutacji na problem eil101



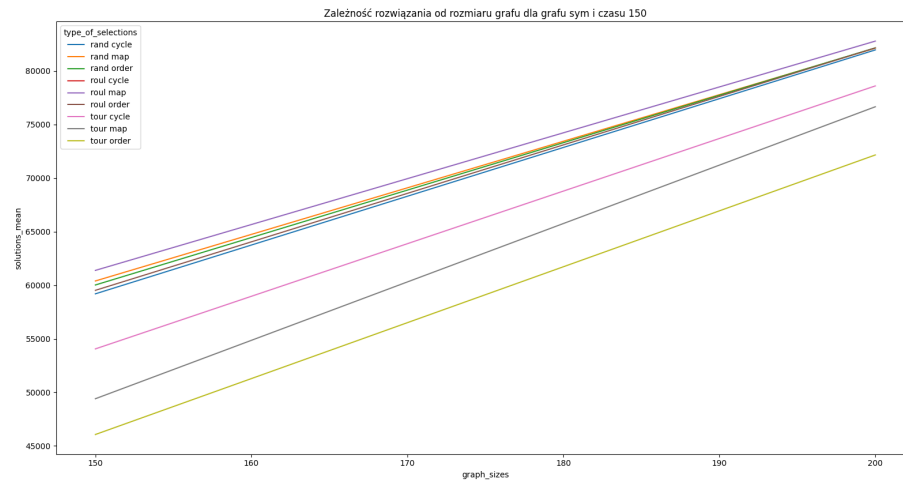
Rysunek 8: Działanie rozmiaru populacji na problem eil101

Na powyższych wykresach możemy zauważyć, że rodzaj selekcji oraz krzyżowania ma wpływ na rezultat. Wyszło tutaj, iż turniejowa selekcja, promująca elitaryzm, osiąga najlepsze wyniki, natomiast selekcja losowa zawsze radzi sobie

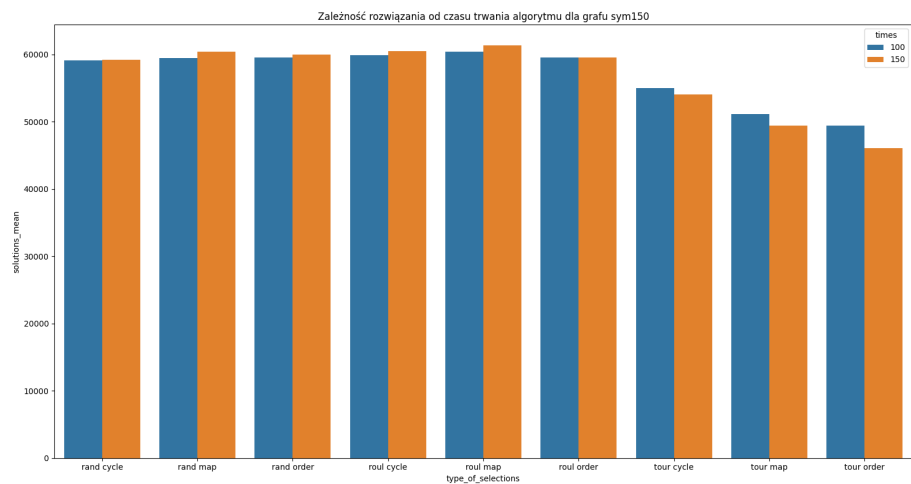
najslabiej. Selekcja poprzez ruletkę, w zależności od rozmiaru grafu, znajduje się bliżej bądź dalej od rozwiązania optymalnego. Jej odbieganie przy większych grafach może być spowodowane zbyt małą liczbą przebiegów - dla takiej stałej próbki czasowej, mogło to być niewystarczające dla osiągnięcia przez ten algorytm maksimum swoich możliwości.

7 Nasza implementacja

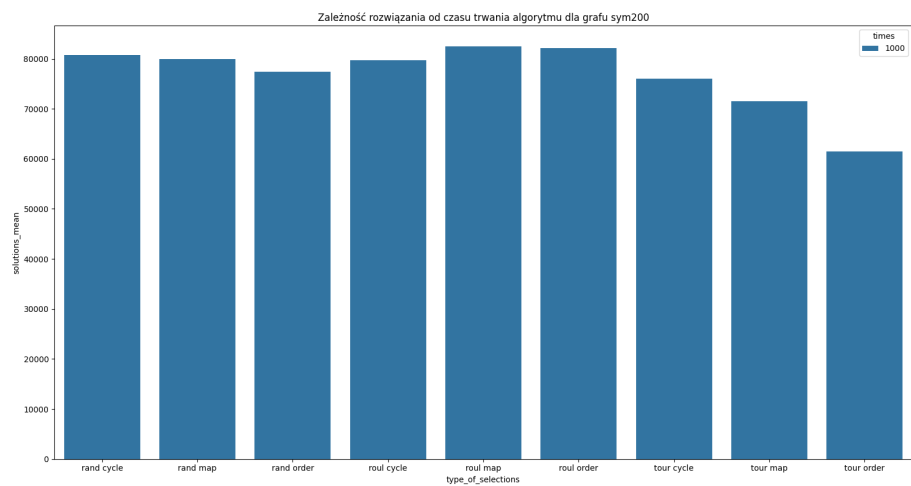
Działania naszego algorytmu na losowym grafie:



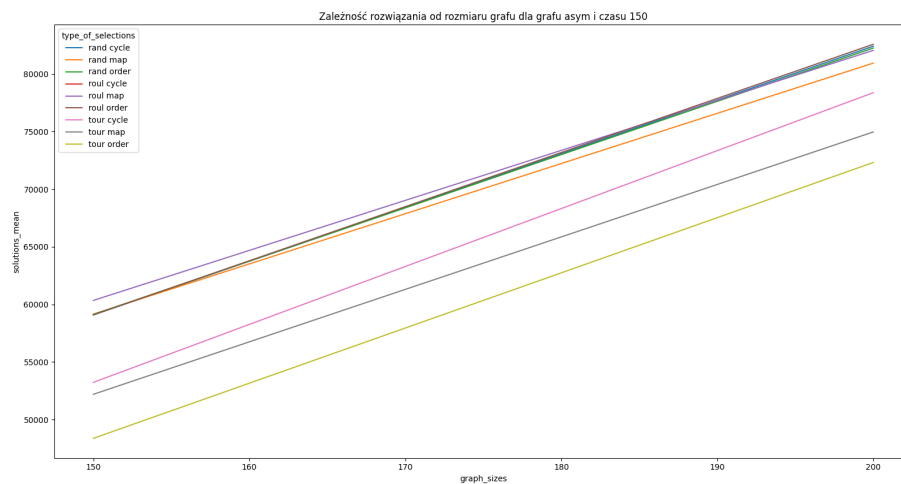
Rysunek 9: Średnia wyników dla czasu 150s



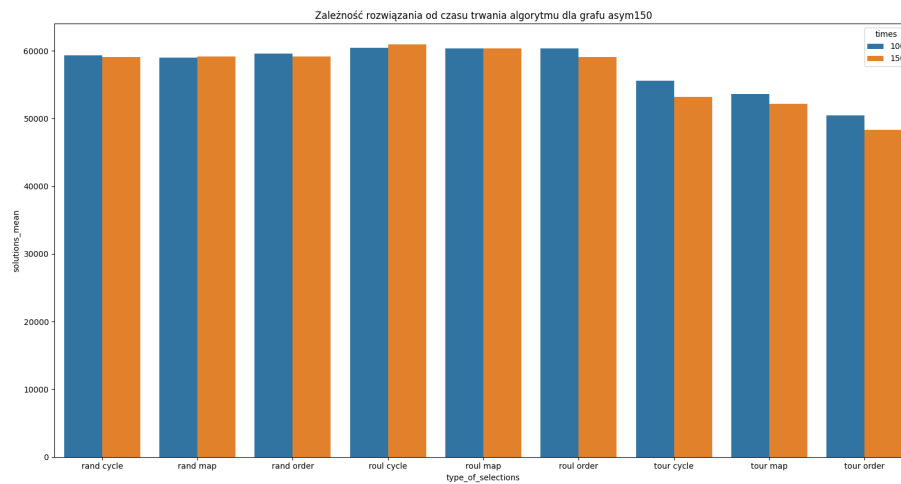
Rysunek 10: Histogram średniej wyników dla czasu 150s



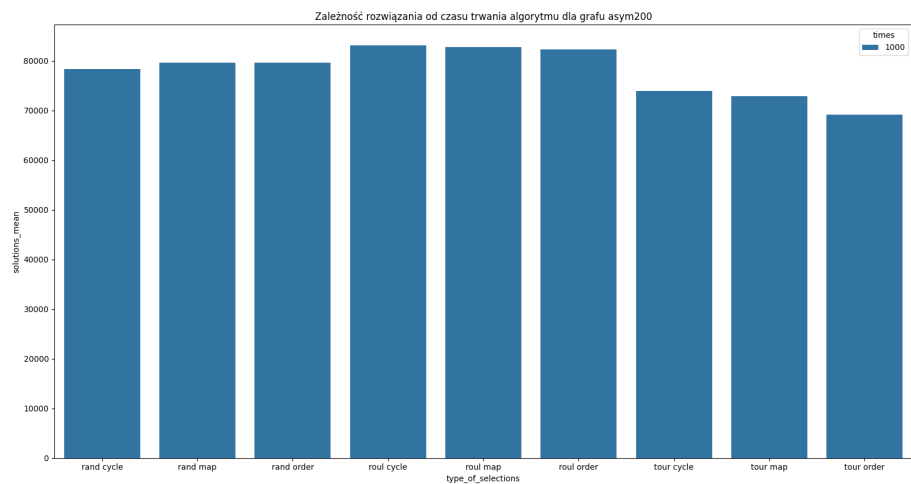
Rysunek 11: Srednia wyników dla garfu 200 i czasu 1000s



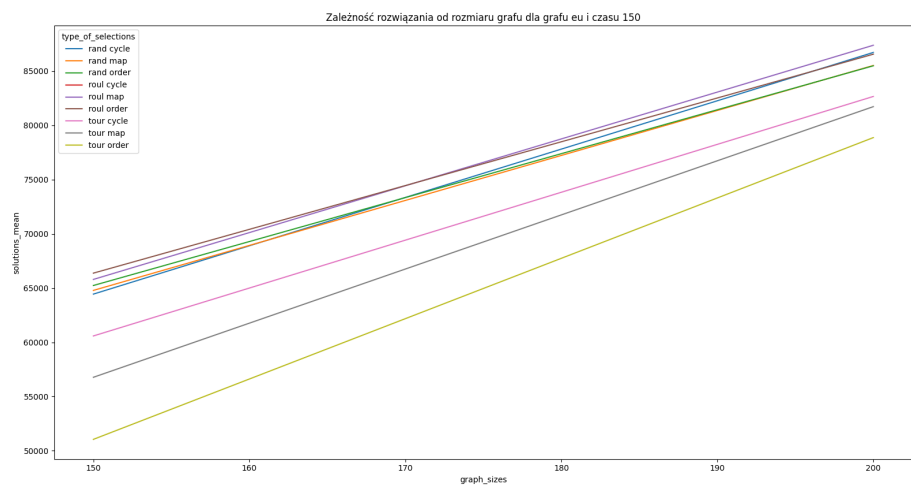
Rysunek 12: Średnia wyników dla czasu 150s



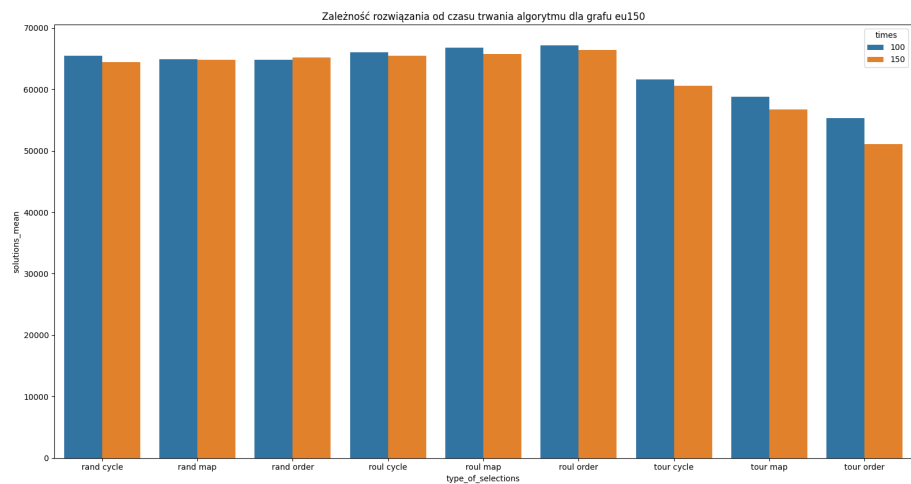
Rysunek 13: Histogram średniej wyników dla czasu 150s



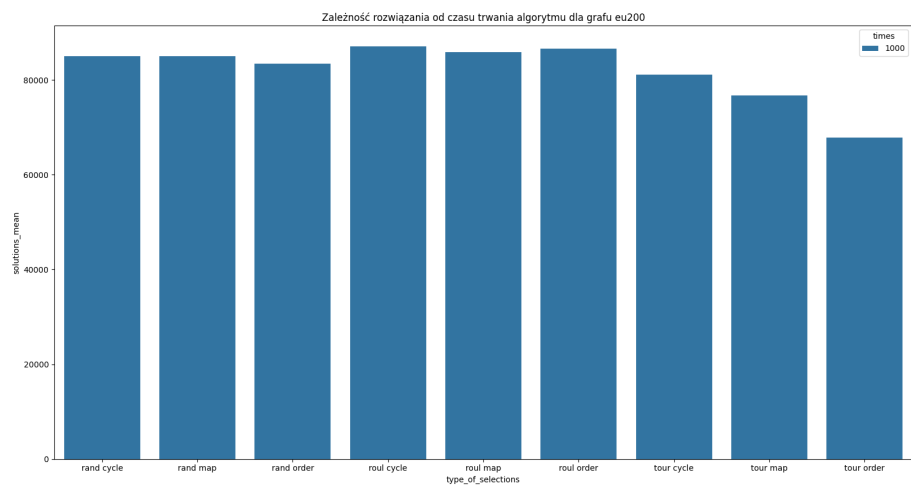
Rysunek 14: Średnia wyników dla grafu 200 i czasu 1000s



Rysunek 15: Średnia wyników dla czasu 150s



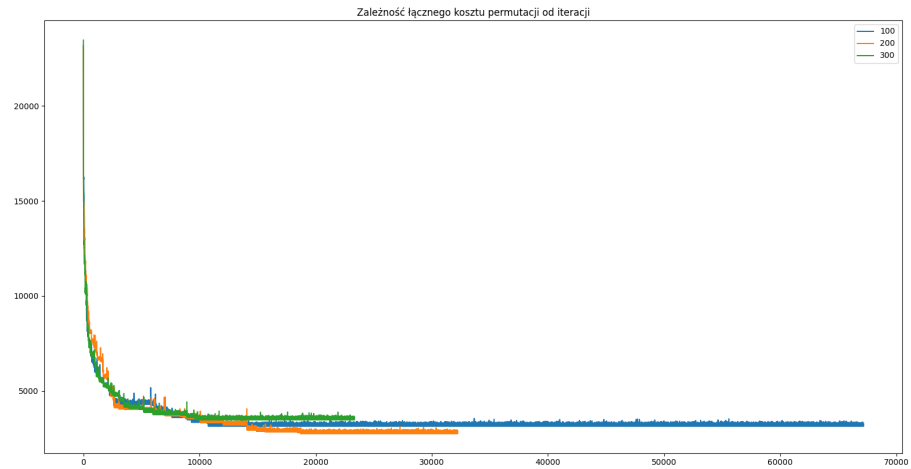
Rysunek 16: Histogram średniej wyników dla czasu 150s



Rysunek 17: Srednia wyników dla garfu 200 i czasu 1000s

8 Test populacji

Przeprowadziliśmy również te testy, które pozwalają nam oszacować wpływ wielkości początkowej na zbieżność rozwiązania.



Rysunek 18: Wpływ populacji na znajdowanie optymalnego rozwiązania