

**AKADEMIA GÓRNICZO – HUTNICZA   
 im. Stanisława Staszica w Krakowie**

WYDZIAŁ ZARZĄDZANIA

Inteligencja Obliczeniowa

Temat projektu:

*Implementacja i analiza algorytmów Wspinaczki, Symulowanego Wyżarzania, Przeszukiwania Tabu, Najbliższego Sąsiada oraz Algorytmu Genetycznego.*

Imię i nazwisko autorów: Michał Wójcik, Karol Kuciński, Bartek Zajdel,

Kierunek: Informatyka i ekonometria  
Rok studiów: 2022, studia stacjonarne

# Algorytm Wspinaczki

Algorytm wspinaczki to heurystyka stosowana w sztucznej inteligencji do rozwiązywania problemów optymalizacyjnych. Jego ideą jest stopniowe zmierzanie do rozwiązania poprzez ciągłe poprawianie bieżącego rozwiązania, tak aby zbliżać się do optymalnego. Algorytm działa poprzez iteracyjne zmienianie rozwiązania i porównywanie go z poprzednim, w celu znalezienia lepszego rozwiązania. Jeśli zmiana prowadzi do lepszego rozwiązania, algorytm ją akceptuje i przechodzi do kolejnej iteracji. W przeciwnym razie, algorytm zatrzymuje się i zwraca bieżące rozwiązanie jako najlepsze znalezione do tej pory.

Podejście to jest dość proste, ale ma pewne ograniczenia. Algorytm może zatrzymać się w lokalnym, a nie globalnym minimum lub maksimum, jeśli istnieje kilka lokalnych punktów optymalnych w przestrzeni rozwiązań. W takim przypadku, algorytm wspinaczki może nie być w stanie znaleźć globalnie optymalnego rozwiązania. Zauważono że wyniki dla małej liczby sąsiadów są niezadawalające dlatego w jednym kroku algorytmu liczymy wszystkich możliwych sąsiadów dla danego rozwiązania a w następnym generujemy konkretną liczbę tych sąsiadów.

*Badanie wpływu rozmiaru sąsiedztwa przy stałej liczbie startów*

Dane\_TSP\_48

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rozmiar sąsiedztwa | 5 | 10 | 25 | 40 |
| Liczba startów | 6 | 6 | 6 | 6 |
| Wynik | 12697 | 14580 | 14887 | 14542 |

Dane\_TSP\_76

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rozmiar sąsiedztwa | 5 | 10 | 25 | 50 |
| Liczba startów | 4 | 4 | 4 | 4 |
| Wynik | 161581.6426448 | 160645.20908850004 | 148936.4393056 | 158021.26172250003 |

Dane\_TSP\_127

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rozmiar sąsiedztwa | 5 | 10 | 25 | 60 |
| Liczba startów | 6 | 6 | 6 | 6 |
| Wynik | 187934.16164920005 | 165532.97457629998 | 179627.3550807 | 195717.2612722 |

Rozmiar sąsiedztwa ma duży wpływ na wyniki algorytmów. Jeśli rozmiar sąsiedztwa jest duży, algorytm ma większe szanse na znalezienie lepszego rozwiązania, ale jednocześnie jest bardziej czasochłonny i może być mniej efektywny.

Natomiast jeśli rozmiar sąsiedztwa jest mały, algorytm jest szybszy, ale ma mniejsze szanse na znalezienie lepszego rozwiązania. Dlatego ważne jest, aby dobrać odpowiedni rozmiar sąsiedztwa w zależności od konkretnego problemu i oczekiwanej skuteczności algorytmu. Na podstawie analizy podanych wyników ciężko wyciągnąć konkretne wnioski. W większości przypadków zwiększając rozmiar sąsiedztwa wynik staję się lepszy. Występują jednak przypadki w których mimo zwiększenia sąsiedztwa wynik jest gorszy co może być spowodowane dużą losowością programu, program za każdym razem generuje początkowe losowe rozwiązanie które ma duży wpływ na dalsze obliczenia.

*Badanie wpływu liczby startów na wynik przy stałym sąsiedztwie. W przeprowadzenia zależności wybrano parametry które w poprzednim badaniu uzyskały najlepsze wyniki.*

Dane\_TSP\_48

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Liczba startów | 3 | 5 | 7 | 9 |
| Rozmiar sąsiedztwa | 25 | 25 | 25 | 25 |
| Wynik | 2307 | 13557 | 14016 | 13952 |

Dane\_TSP\_76

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Liczba startów | 3 | 5 | 7 | 9 |
| Rozmiar sąsiedztwa | 25 | 25 | 25 | 25 |
| Wynik | 160155.25000149995 | 155691.72359310003 | 178524.81976520002 | 168434.4857749 |

Dane\_TSP\_127

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Liczba startów | 3 | 5 | 7 | 9 |
| Rozmiar sąsiedztwa | 25 | 25 | 25 | 25 |
| Wynik | 193499.53445940002 | 172591.53529980002 | 181147.82011410003 | 173119.4954524 |

Liczba startów istotnie wpływa na znalezienia optymalnego rozwiązania. Im więcej startów tym lepiej, ponieważ przeszukujemy większą liczbę możliwych optymalnych rozwiązań. Algorytm rozpoczyna się poprzez generowanie losowego rozwiązania dlatego wyniki w powyższych tabelkach pomimo zwiększenia liczby iteracji nie zawsze są lepsze.

*Badanie jak wpływa metoda doboru sąsiadów na najlepsze rozwiązania*

Generacja sąsiadów przyprowadzona była za pomocą metody „SWAP”. Postanowiono zbadać czy druga metoda generowania sąsiadów która być może spowoduje lepsze wyniki. Dla parametrów które uzyskały najlepsze wynik z każdego Excela zastosowano metodę „INVERSION” dobierania sąsiadów. Metoda „INVERSION” polega na tym, że dla dwóch losowo wybranych miast zamienia się kolejność miast między nimi. W ten sposób generuje się nowe rozwiązanie, które jest sąsiadem oryginalnego rozwiązania

Dane\_TSP\_48

|  |  |  |
| --- | --- | --- |
|  | SWAP | INVERSION |
| Liczba startów | 6 | 6 |
| Rozmiar sąsiedztwa | 5 | 5, 48 ( w drugim kroku) |
| Wynik | 1259112697 | 15897 |

Dane\_TSP\_76

|  |  |  |
| --- | --- | --- |
|  | SWAP | INVERSION |
| Liczba startów | 4 | 4 |
| Rozmiar sąsiedztwa | 25 | 25 , 76( w drugim kroku) |
| Wynik | 148936.4393056 | 182716.2493398 |

Dane\_TSP\_127

|  |  |  |
| --- | --- | --- |
|  | SWAP | INVERSION |
| Liczba startów | 10 | 10 |
| Rozmiar sąsiedztwa | 10 | 10, 127( w drugim kroku) |
| Wynik | 165532.97457629998 | 191205.85031349995 |

Na podstawie powyższych tabeli możemy wyciągnąć wnioski że metoda „SWAP” w przypadku naszych konkretnych danych i ich parametrów działa lepiej niż metoda „INVERSION”.

# Symulowane Wyżarzanie

Jest metoda optymalizacji, która polega na stopniowym obniżaniu temperatury w trakcie wirtualnego procesu termicznego, podczas którego algorytm przeszukuje różne możliwe rozwiązania danego problemu i wybiera te, które prowadzą do najlepszego rozwiązania. W ten sposób symulowane wyżarzanie pozwala na znalezienie optymalnego rozwiązania nawet wtedy, gdy istnieje duże pole możliwych rozwiązań. Algorytm działa poprzez iteracyjne losowanie nowych rozwiązań, obliczanie ich energii oraz porównywanie ich z aktualnym rozwiązaniem. Nowe rozwiązanie jest akceptowane z pewnym prawdopodobieństwem, które zależy od różnicy energii między rozwiązaniami oraz od aktualnej temperatury. Po każdej iteracji temperatura jest stopniowo obniżana, co zmniejsza prawdopodobieństwo akceptacji gorszych rozwiązań. Algorytm działa aż do osiągnięcia pożądanej temperatury lub aż do osiągnięcia maksymalnej liczby iteracji, po czym zwraca najlepsze znalezione rozwiązanie.

*Badanie wpływu temperatury początkowej przy stałej temperaturze końcowej, wartości alfa oraz przy takiej samej liczbie iteracji dla danego arkusza*

Dane\_TSP\_48

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Temperatura początkowa | 10 | 50 | 100 | 500 |
| Temperatura  końcowa | 0.1 | 0.1 | 0.1 | 0.1 |
| Liczba iteracji | 5 | 5 | 5 | 5 |
| Wartość alfa | 0.99 | 0.99 | 0.99 | 0.99 |
| Wynik | 23740 | 21302 | 20024 | 21477 |

Dane\_TSP\_76

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Temperatura początkowa | 10 | 50 | 100 | 500 |
| Temperatura  końcowa | 0.1 | 0.1 | 0.1 | 0.1 |
| Liczba iteracji | 10 | 10 | 10 | 10 |
| Wartość alfa | 0.99 | 0.99 | 0.99 | 0.99 |
| Wynik | 293850.5953974999 | 325962.12182100007 | 282379.4313753998 | 271123.0927241 |

Dane\_TSP\_127

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Temperatura początkowa | 10 | 50 | 100 | 500 |
| Temperatura  końcowa | 0.1 | 0.1 | 0.1 | 0.1 |
| Liczba iteracji | 20 | 20 | 20 | 20 |
| Wartość alfa | 0.99 | 0.99 | 0.99 | 0.99 |
| Wynik | 431596.0204092001 | 383303.50654850015 | 366082.3913237998 | 360204.2352595 |

W większości przypadków gdy temperatura początkowa jest zwiększana wynik się polepsza. Przypadki w których pomimo zwiększonej temperatury wynik jest gorszy mogą być spowodowane dużą losowością, za każdym razem losujemy pierwsze rozwiązanie które ma w kolejnych etap duży wpływ na rozwiązanie problemu. Temperatura początkowa ma istotny wpływ na wyniki. Na podstawie obliczonych tabelek można wyciągnąć wnioski że im wyższa temperatura początkowa tym lepszy wyniki który jest jednocześnie dłużej obliczany. Mimo że wyniki nie są zadawalające to poprzez zmienianie parametrów stają się one coraz lepsze. W kolejnym badaniu zostanie wykorzystana temperatura początkowa równa 100 oraz 500 w zależności od odpowiedniego arkusza Excela. Są to temperatury początkowe, dla których algorytm pokazał najlepsze wyniki.

*Badanie wpływu temperatury końcowej przy stałej temperaturze początkowej oraz przy takiej samej liczbie sąsiedztwa, iteracji dla danego arkusza*

Dane\_TSP\_48

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Temperatura  końcowa | 0,5 | 0,1 | 0,05 | 0,02 |
| Temperatura początkowa | 100 | 100 | 100 | 100 |
| Liczba iteracji | 20 | 20 | 20 | 20 |
| Wynik | 20821 | 19416 | 22162 | 20389 |

Dane\_TSP\_76

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Temperatura  końcowa | 0,5 | 0,1 | 0,05 | 0,02 |
| Temperatura początkowa | 500 | 500 | 500 | 500 |
| Liczba iteracji | 20 | 20 | 20 | 20 |
| Wynik | 300709.5192071999 | 312368.0429733999 | 269023.9613012 | 254929.39947019998 |

Dane\_TSP\_127

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Temperatura  końcowa | 0,5 | 0,1 | 0,05 | 0,02 |
| Temperatura początkowa | 500 | 500 | 500 | 500 |
| Liczba iteracji | 20 | 20 | 20 | 20 |
| Wynik | 373722.53181350016 | 353449.16731519985 | 340796.5320400999 | 330665.61525170004 |

Temperatura końcowa jest jednym z kluczowych parametrów dla algorytmu symulowanego wyżarzania. Jeśli temperatura końcowa jest zbyt wysoka, algorytm może zbyt często akceptować zmiany, co może prowadzić do nieoptymalnych rozwiązań. Jeśli temperatura końcowa jest zbyt niska, algorytm może zbyt rzadko akceptować zmiany, co może prowadzić do zatrzymania się w lokalnym minimum. Na podstawie powyższych tabeli zmniejszając temperaturę końcową praktycznie za każdym razem otrzymujemy coraz to lepsze wyniki. W kolejnych badaniach wykorzystana zostanie temperatura końcowa równa 0,02 dla której algorytm generował najlepsze wyniki.

*Badanie wpływu liczby sąsiadów*

Dane\_TSP\_48

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Liczba sąsiadów | 10 | 15 | 20 | 25 |
| Temperatura  końcowa | 0,02 | 0,02 | 0,02 | 0,02 |
| Temperatura początkowa | 500 | 500 | 500 | 500 |
| Liczba iteracji | 20 | 20 | 20 | 20 |
| Wynik | 37954 | 31407 | 32715 | 28453 |

Dane\_TSP\_76

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Liczba sąsiadów | 10 | 15 | 20 | 25 |
| Temperatura  końcowa | 0,02 | 0,02 | 0,02 | 0,02 |
| Temperatura początkowa | 500 | 500 | 500 | 500 |
| Liczba iteracji | 20 | 20 | 20 | 20 |
| Wynik | 481288.5331742001 | 484186.5202677001 | 452903.6607764002 | 467638.9469201999 |

Dane\_TSP\_127

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Liczba sąsiadów | 10 | 15 | 20 | 25 |
| Temperatura  końcowa | 0,02 | 0,02 | 0,02 | 0,02 |
| Temperatura początkowa | 500 | 500 | 500 | 500 |
| Liczba iteracji | 20 | 20 | 20 | 20 |
| Wynik | 547758.8049596999 | 530232.6249907002 | 580762.6831679 | 548408.9491753995 |

Liczba sąsiedztwa określa, ile zmian będzie dokonywane w każdym kroku algorytmu. Im większa liczba sąsiedztwa, tym więcej zmian będzie wprowadzanych w każdym kroku, co oznacza, że algorytm będzie bardziej skomplikowany, ale może również prowadzić do lepszych rozwiązań. Im mniejsza liczba sąsiedztwa, tym mniej zmian będzie wprowadzanych w każdym kroku, co oznacza, że algorytm będzie prostszy, ale może również prowadzić do gorszych rozwiązań. Ciężko wyciągnąć konkretne wnioski na podstawie obliczonych wyników, występują zarówno sytuacje w których wyniki są lepsze oraz gorsze w związku z zwiększaniem liczby sąsiadów, zapewne jest to spowodowane kilkoma czynnikami takimi jak niekoniecznie dobrze dobrane parametry czy duża losowość.

# Algorytm najbliższego sąsiada

Algorytm najbliższego sąsiada – algorytm służący do rozwiązywania problemu komiwojażera. Jest to algorytm wykorzystujący strategię zachłanną.

Działanie algorytmu

Algorytm rozpoczyna działanie od wybranego wierzchołka i polega na kolejnym przechodzeniu do najbliższego nieodwiedzonego sąsiada ostatnio dodanego wierzchołka. W bardziej formalnym zapisie algorytm działa w następujący sposób:

1. Oznaczenie wierzchołka początkowego jako odwiedzony i ustawienie go jako aktualny.
2. Wybranie najkrótszej spośród krawędzi łączących aktualny wierzchołek z jeszcze nieodwiedzonymi wierzchołkami.
3. Dołączenie do rozwiązania krawędzi znalezionej w punkcie 2.
4. Oznaczenie wierzchołka będącego drugim końcem krawędzi znalezionej w punkcie 2 jako odwiedzony i ustawienie jako aktualny.
5. Jeśli są jeszcze nieodwiedzone wierzchołki, przejście do punktu 2.
6. Dołączenie krawędzi łączącej ostatnio dodany wierzchołek z wierzchołkiem początkowym.

TSP\_29

|  |  |  |
| --- | --- | --- |
| Punkt startowy | Trasa | Dystans |
| 3 | [3.0 29.0 26.0 5.0 9.0 6.0 12.0 28.0 1.0 21.0 2.0 20.0 10.0 4.0 15.0 19.0 16.0 27.0 24.0 8.0 23.0 25.0 7.0 11.0 22.0 14.0 18.0 17.0 13.0] | 2386.0 |
| 6 | [6.0 12.0 9.0 5.0 26.0 29.0 3.0 2.0 21.0 1.0 28.0 8.0 27.0 24.0 16.0 19.0 15.0 4.0 10.0 20.0 13.0 25.0 7.0 23.0 11.0 22.0 14.0 18.0 17.0] | 2507.0 |
| 9 | [9.0 5.0 26.0 29.0 3.0 2.0 21.0 1.0 28.0 6.0 12.0 8.0 27.0 24.0 16.0 19.0 15.0 4.0 10.0 20.0 13.0 25.0 7.0 23.0 11.0 22.0 14.0 18.0 17.0] | 2588.0 |
| 24 | [24.0 27.0 8.0 1.0 28.0 6.0 12.0 9.0 5.0 26.0 29.0 3.0 2.0 21.0 20.0 10.0 4.0 15.0 19.0 16.0 13.0 25.0 7.0 23.0 11.0 22.0 14.0 18.0 17.0] | 2438.0 |

Dane\_TSP\_48

|  |  |  |
| --- | --- | --- |
| Punkt startowy | Trasa | Dystans |
| 4 | [4.0 26.0 10.0 24.0 42.0 48.0 5.0 29.0 2.0 34.0 14.0 25.0 13.0 23.0 11.0 12.0 15.0 33.0 46.0 44.0 18.0 7.0 28.0 36.0 30.0 6.0 37.0 19.0 27.0 43.0 17.0 20.0 47.0  21.0 39.0 32.0 3.0 22.0 16.0 1.0 9.0 38.0 31.0 40.0 8.0 41.0 35.0 45.0] | 13987.0 |
| 14 | [14.0 25.0 13.0 23.0 11.0 12.0 15.0 33.0 46.0 44.0 18.0 7.0 28.0 36.0 30.0 6.0 37.0 19.0 27.0 43.0 17.0 20.0 47.0 21.0 39.0 32.0 48.0 5.0 29.0 2.0 42.0 10.0 24.0 45.0 35.0 26.0 4.0 34.0 3.0 22.0 16.0 1.0 9.0 38.0 31.0 40.0 8.0 41.0] | 12956.0 |
| 24 | [24.0 10.0 42.0 48.0 5.0 29.0 2.0 26.0 4.0 35.0 45.0 32.0 39.0 25.0 14.0 23.0 11.0 12.0 15.0 33.0 46.0 44.0 18.0 7.0 28.0 36.0 30.0 6.0 37.0 19.0 27.0 43.0 17.0 20.0 47.0 21.0 13.0 34.0 3.0 22.0 16.0 1.0 9.0 38.0 31.0 40.0 8.0 41.0] | 13369.0 |
| 34 | [34.0 14.0 25.0 13.0 23.0 11.0 12.0 15.0 33.0 46.0 44.0 18.0 7.0 28.0 36.0 30.0 6.0 37.0 19.0 27.0 43.0 17.0 20.0 47.0 21.0 39.0 32.0 48.0 5.0 29.0 2.0 42.0 10.0 24.0 45.0 35.0 26.0 4.0 41.0 16.0 22.0 3.0 40.0 9.0 1.0 8.0 38.0 31.0] | 12731.0 |

Dane\_TSP\_76

|  |  |  |
| --- | --- | --- |
| Punkt startowy | Trasa | Dystans |
| 11 | [11.0 12.0 13.0 14.0 15.0 16.0 17.0 18.0 37.0 36.0 35.0 34.0 40.0 41.0 60.0 59.0 58.0 57.0 63.0 64.0 62.0 61.0 55.0 56.0 51.0 66.0 65.0 50.0 49.0 52.0 53.0 54.0 42.0 43.0 28.0 29.0 30.0 31.0 19.0 20.0 5.0 6.0 7.0 8.0 9.0 10.0 4.0 3.0 2.0 1.0 23.0 22.0 21.0 25.0 24.0 46.0 45.0 44.0 48.0 47.0 69.0 68.0 67.0 70.0 71.0 72.0 73.0 39.0 38.0 32.0 33.0 27.0 26.0 75.0 76.0 74.0] | 131607.48 |
| 31 | [31.0 19.0 20.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 16.0 17.0 18.0 37.0 36.0 35.0 34.0 40.0 41.0 60.0 59.0 58.0 57.0 63.0 64.0 62.0 61.0 55.0 56.0 51.0 66.0 65.0 50.0 49.0 52.0 53.0 54.0 42.0 43.0 28.0 29.0 30.0 26.0 27.0 33.0 32.0 38.0 39.0 4.0 3.0 2.0 1.0 23.0 22.0 21.0 25.0 24.0 46.0 45.0 44.0 48.0 47.0 69.0 68.0 67.0 70.0 71.0 72.0 73.0 74.0 75.0 76.0] | 142610.54 |
| 51 | [51.0 66.0 65.0 56.0 55.0 52.0 53.0 54.0 42.0 43.0 28.0 29.0 30.0 31.0 19.0 20.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 16.0 17.0 18.0 37.0 36.0 35.0 34.0 40.0 41.0 60.0 59.0 58.0 57.0 63.0 64.0 62.0 61.0 39.0 38.0 32.0 33.0 27.0 26.0 25.0 21.0 22.0 23.0 1.0 2.0 3.0 4.0 24.0 46.0 45.0 44.0 48.0 47.0 69.0 68.0 67.0 50.0 49.0 71.0 72.0 73.0 70.0 76.0 75.0 74.0] | 153822.66 |
| 71 | [71.0 72.0 73.0 64.0 63.0 57.0 58.0 59.0 60.0 41.0 40.0 34.0 35.0 36.0 37.0 18.0 17.0 11.0 12.0 13.0 14.0 15.0 16.0 10.0 9.0 6.0 7.0 8.0 3.0 4.0 5.0 20.0 19.0 31.0 30.0 29.0 28.0 43.0 42.0 54.0 53.0 52.0 51.0 66.0 65.0 56.0 55.0 49.0 50.0 67.0 68.0 69.0 47.0 48.0 44.0 45.0 46.0 24.0 25.0 21.0 22.0 23.0 1.0 2.0 75.0 76.0 26.0 27.0 33.0 32.0 38.0 39.0 61.0 62.0 70.0 74.0] | 154936.761 |

Dane\_TSP\_127

|  |  |  |
| --- | --- | --- |
| Punkt startowy | Trasa | Dystans |
| 9 | [9.0 11.0 114.0 105.0 106.0 6.0 108.0 24.0 23.0 4.0 22.0 21.0 17.0 20.0 15.0 19.0 72.0 8.0 67.0 73.0 74.0 18.0 77.0 79.0 80.0 31.0 27.0 30.0 41.0 14.0 12.0 16.0 1.0 7.0 120.0 13.0 115.0 50.0 51.0 2.0 37.0 36.0 35.0 40.0 43.0 34.0 39.0 42.0 38.0 26.0 25.0 33.0 122.0 28.0 29.0 32.0 117.0 84.0 81.0 126.0 82.0 83.0 75.0 76.0 78.0 68.0 71.0 70.0 69.0 96.0 109.0 87.0 88.0 86.0 85.0 110.0 59.0 62.0 61.0 91.0 90.0 116.0 60.0 3.0 10.0 100.0 58.0 64.0 113.0 66.0 55.0 124.0 52.0 5.0 121.0 56.0 57.0 54.0 45.0 103.0 44.0 48.0 118.0 53.0 49.0 47.0 46.0 94.0 112.0 111.0 107.0 127.0 93.0 95.0 123.0 97.0 98.0 101.0 102.0 63.0 119.0 104.0 125.0 89.0 92.0 99.0 65.0] | 134168.26 |
| 39 | [39.0 34.0 43.0 40.0 35.0 37.0 36.0 41.0 14.0 12.0 31.0 27.0 30.0 26.0 38.0 42.0 25.0 33.0 122.0 28.0 29.0 32.0 80.0 79.0 77.0 18.0 21.0 17.0 22.0 4.0 23.0 24.0 108.0 6.0 106.0 15.0 105.0 114.0 11.0 9.0 8.0 72.0 19.0 20.0 7.0 120.0 13.0 115.0 50.0 51.0 2.0 16.0 1.0 10.0 3.0 90.0 116.0 60.0 62.0 61.0 91.0 58.0 64.0 100.0 52.0 124.0 56.0 5.0 121.0 57.0 54.0 45.0 103.0 44.0 48.0 118.0 53.0 49.0 47.0 46.0 94.0 112.0 111.0 107.0 127.0 93.0 95.0 123.0 97.0 117.0 84.0 81.0 126.0 82.0 83.0 75.0 76.0 78.0 74.0 73.0 67.0 68.0 71.0 70.0 69.0 96.0 109.0 87.0 88.0 86.0 85.0 110.0 59.0 125.0 89.0 92.0 99.0 65.0 113.0 66.0 55.0 104.0 119.0 63.0 102.0 101.0 98.0] | 145591.27 |
| 68 | [68.0 73.0 74.0 18.0 77.0 79.0 80.0 31.0 27.0 30.0 41.0 14.0 12.0 20.0 108.0 6.0 106.0 15.0 105.0 114.0 11.0 9.0 24.0 23.0 4.0 22.0 21.0 17.0 19.0 72.0 8.0 67.0 71.0 70.0 69.0 75.0 76.0 78.0 117.0 84.0 81.0 126.0 82.0 83.0 32.0 29.0 33.0 122.0 28.0 25.0 26.0 38.0 39.0 34.0 43.0 40.0 35.0 37.0 36.0 16.0 1.0 7.0 120.0 13.0 115.0 50.0 51.0 2.0 121.0 5.0 56.0 124.0 52.0 55.0 66.0 113.0 64.0 58.0 100.0 10.0 3.0 90.0 116.0 60.0 62.0 61.0 91.0 59.0 125.0 89.0 92.0 99.0 65.0 47.0 53.0 49.0 118.0 48.0 46.0 94.0 112.0 111.0 107.0 127.0 93.0 95.0 123.0 97.0 42.0 44.0 45.0 103.0 54.0 57.0 110.0 109.0 96.0 88.0 87.0 86.0 85.0 119.0 63.0 102.0 101.0 98.0 104.0] | 149097.50 |
| 99 | [99.0 92.0 89.0 125.0 113.0 64.0 58.0 100.0 10.0 120.0 13.0 115.0 50.0 51.0 2.0 16.0 1.0 7.0 105.0 114.0 106.0 6.0 108.0 24.0 23.0 4.0 22.0 21.0 17.0 20.0 15.0 11.0 9.0 8.0 72.0 19.0 18.0 77.0 74.0 73.0 67.0 68.0 71.0 70.0 69.0 75.0 76.0 78.0 117.0 84.0 81.0 126.0 82.0 83.0 32.0 29.0 33.0 122.0 28.0 25.0 26.0 38.0 39.0 34.0 43.0 40.0 35.0 37.0 36.0 41.0 14.0 12.0 31.0 27.0 30.0 42.0 44.0 45.0 103.0 54.0 57.0 121.0 5.0 56.0 124.0 52.0 55.0 66.0 47.0 53.0 49.0 118.0 48.0 46.0 94.0 112.0 111.0 107.0 127.0 93.0 95.0 123.0 97.0 80.0 79.0 3.0 90.0 116.0 60.0 62.0 61.0 91.0 59.0 110.0 109.0 96.0 88.0 87.0 86.0 85.0 119.0 63.0 102.0 101.0 98.0 65.0 104.0] | 144608.32 |

Algorytm genetyczny

Jest to algorytm działający na zasadzie genetyki. Początkowo losuje się pewną ilość rozwiązań nazywanych populacją a następnie z danej populacji losowana jest próba. Na podstawie tej próby i jakiejś wcześniej określonej fitness function (funkcji dopasowania ) algorytm znajduje najlepsze osobniki ( pojedyńczy element populacji np. Jedno rozwiązanie ) które następnie oznacza się jako mating\_pool czyli przestrzeń doboru rodziców. Następnie za pomocą technik Fitness proportionate selection bądź Tournament selection wybiera się n par rodziców. Osobniki te za pomocą technik krzyżowania dają nowe pary potomków co daje nowe pokolenie. Nowe pokolenie jest następnie przyjmowane za populację i kroki są powtarzane do momentu gdy ilość pokoleń nie zostanie przekroczona.

Przypadek 48 miast

Pierwszym testowanym parametrem był dobór rodziców w kombinacji z metodą krzyżowania rodziców w celu uzyskania potomków. Jak widać najlepiej poradziła sobie kombinacja krzyżowania MkX oraz Fitness proportionate selection.

Następnym parametrem był rozmiar populacji. Nie dziwnym jest wynik sugerujący, że im więcej mamy osobników, tym lepszych rodziców możemy dobrać.

Podobne zachowanie wystąpiło w wypadku liczby potomków.

Najlepszym prawdopodobieństwem mutacji okazało się 0.2. Sugeruje to, że nadal powinno być niskie tj. <0.3 lecz nadal znaczące. Szansa 1:5 zadziałała w tym wypadku najlepiej.

Kolejny wynik nie budzi konsternacji. Im więcej pokoleń (=iteracji alg. Genetycznego) tym lepsze finalnie wyniki uzyskujemy.

A picture containing timeline

Description automatically generated

TSP 76

W wypadku nieco większego problemu lepiej zadziałało tournament selection, nadal z krzyżowaniem MkX.

Tu zaś zbyt duża populacja nie pomogła zbyt wiele i dla większej ilości danych najlepiej sobię poradził dla w miarę niskiej wartości tego parametru.

Podobnie jak w poprzednim problemie im większa liczba potomków tym lepsze rezultaty.

Tutaj również prawdopodobieństwo niskie ale relatywnie znaczące = 0.2 poradziło sobie najlepiej.

Tak jak w poprzednim przykładzie, największa liczba pokoleń generuje najlepsze rezultaty.

Calendar

Description automatically generated

Przypadek 127 miast

Podobnie jak wcześniej algorytm najlepiej zadziałał dla metody MkX oraz Tournament Selection.

Zaskakującym jest fakt, że najlepszy rezultat uzyskano dla najmniejszej wartości parametru populacja.

Ilość potomków w tym wypadku najlepiej działała gdy była dość duża.

W wypadku tak dużego problemu najlepsze rezultaty dało prawdopodobieństwo równe 0.1

Niezmiennym wnioskiem jest fakt, że najlepsze rezultaty daje większa ilość pokoleń.

Calendar

Description automatically generated

Podsumowanie GA

Najlepszą metodą krzyżowania bez warunkowo jest metoda MkX. Na małej próbce ilości problemów ryzyko błędnych konkluzji jest spore jednak uważam, że można założyć iż Tournament Selection jest lepszą metodą w wypadku dużych problemów. Prawdopodobieństwo mutacji należy ustawiać na w miarę niskim poziomie a lepsze rezultaty na pewno zagwarantuje nam większa liczba pokoleń.

Taboo Search

Taboo search jest metaheurystyką służącą do optymalizacji problemów. Jak każda metaheurystyka znajduje rozwiązania optymalne, a nie rozwiązania najbardziej optymalne. Polega on na znajdowaniu rozwiązań I użycie tak zwanych ruchów taboo które są zabronione przez n liczby iteracji. Dzięki temu algorytm może wyjść z minimum lokalnego I zacząć przeszukiwać w stronę minimum globalnego.

TSP 48

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Iteracje | 300 | 500 | 1000 |  | 2000 |
| Długość listy taboo | 3 | 3 | 3 |  | 3 |
| Wynik | 15809 | 15809 | 15809 |  | 15809 |
| Ścieżka | Any[42, 26, 4, 10, 39, 47, 12, 15, 9, 1, 40, 3, 23, 13, 11, 33, 46, 18, 7, 36, 20, 21, 32, 24, 35, 45, 43, 17, 27, 19, 37, 6, 30, 28, 44, 31, 38, 8, 22, 16, 41, 2, 29, 34, 14, 25, 48, 5] | Any[42, 26, 4, 10, 39, 47, 12, 15, 9, 1, 40, 3, 23, 13, 11, 33, 46, 18, 7, 36, 20, 21, 32, 24, 35, 45, 43, 17, 27, 19, 37, 6, 30, 28, 44, 31, 38, 8, 22, 16, 41, 2, 29, 34, 14, 25, 48, 5] | Any[42, 26, 4, 10, 39, 47, 12, 15, 9, 1, 40, 3, 23, 13, 11, 33, 46, 18, 7, 36, 20, 21, 32, 24, 35, 45, 43, 17, 27, 19, 37, 6, 30, 28, 44, 31, 38, 8, 22, 16, 41, 2, 29, 34, 14, 25, 48, 5] |  | Any[42, 26, 4, 10, 39, 47, 12, 15, 9, 1, 40, 3, 23, 13, 11, 33, 46, 18, 7, 36, 20, 21, 32, 24, 35, 45, 43, 17, 27, 19, 37, 6, 30, 28, 44, 31, 38, 8, 22, 16, 41, 2, 29, 34, 14, 25, 48, 5] |

Już w wypadku 300 iteracji algorytm znajduje najlepsze rozwiązanie. Przy takiej długości listy taboo algorytm nie jest w stanie wyjść z lokalnego minimum.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iteracje | 1000 | 1000 | 1000 | 1000 |
| Długość listy taboo | 3 | 10 | 15 | 25 |
| wynik | 15809 | 15598 | 15471 | 15456 |
| ścieżka | Any[42, 26, 4, 10, 39, 47, 12, 15, 9, 1, 40, 3, 23, 13, 11, 33, 46, 18, 7, 36, 20, 21, 32, 24, 35, 45, 43, 17, 27, 19, 37, 6, 30, 28, 44, 31, 38, 8, 22, 16, 41, 2, 29, 34, 14, 25, 48, 5] | Any[42, 26, 4, 10, 39, 13, 47, 15, 9, 1, 40, 3, 23, 11, 12, 33, 46, 18, 7, 36, 20, 21, 32, 24, 35, 45, 43, 17, 27, 19, 37, 6, 30, 28, 44, 31, 38, 8, 22, 16, 41, 2, 29, 34, 14, 25, 5, 48] | Any[42, 26, 4, 10, 39, 13, 47, 15, 40, 9, 22, 3, 23, 11, 12, 33, 46, 18, 7, 36, 20, 21, 32, 24, 35, 45, 43, 17, 27, 19, 37, 6, 30, 28, 44, 31, 38, 8, 1, 16, 41, 2, 29, 34, 14, 25, 48, 5] | Any[42, 26, 4, 10, 39, 13, 47, 15, 40, 9, 1, 3, 23, 11, 12, 33, 46, 18, 7, 36, 20, 21, 32, 24, 35, 45, 43, 17, 27, 19, 37, 6, 30, 28, 44, 31, 38, 8, 22, 16, 41, 2, 29, 34, 14, 25, 48, 5] |

W wypadku 1000 iteracji algorytm znajduje minimalnie lepsze rozwiązanie przy większej długości listy taboo. W poprzednich przykładach miastami startowymi była kolejność numeryczna 1:48

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iteracje | 1000 | 1000 | 1000 | 1000 |
| Długość listy taboo | 25 | 25 | 25 | 25 |
| Miasto startowe | [38, 3, 19, 22, 21, 47, 17, 25, 15, 13, 12, 26, 4, 28, 7, 24, 10, 34, 29, 32, 42, 27, 36, 43, 35, 39, 11, 33, 40, 23, 30, 48, 16, 20, 14, 37, 9, 41, 44, 18, 6, 45, 31, 2, 1, 8, 46, 5] | 9, 16, 19, 12, 10, 15, 14, 27, 34, 30, 36, 2, 39, 31, 17, 4, 33, 28, 20, 25, 48, 22, 11, 35, 26, 3, 42, 37, 24, 13, 38, 21, 40, 32, 18, 7, 47, 41, 29, 45, 43, 23, 44, 46, 1, 5, 8, 6] | 31, 2, 48, 30, 41, 11, 27, 5, 24, 8, 32, 34, 9, 21, 47, 17, 1, 36, 19, 33, 4, 15, 38, 44, 6, 45, 18, 20, 13, 26, 40, 22, 43, 10, 37, 28, 23, 39, 35, 42, 12, 29, 14, 3, 7, 16, 25, 46] | [40, 20, 18, 9, 29, 48, 27, 10, 23, 45, 46, 24, 25, 41, 12, 35, 21, 19, 47, 13, 22, 34, 37, 30, 7, 28, 16, 4, 2, 6, 39, 42, 26, 17, 15, 14, 36, 5, 31, 38, 32, 1, 33, 43, 11, 3, 8, 44] |
| Wynik | 15020 | 14541 | 13290 | 12571 |
| Ścieżka | Any[8, 9, 38, 31, 7, 28, 6, 30, 21, 32, 24, 26, 4, 35, 45, 10, 42, 2, 29, 37, 19, 17, 27, 43, 33, 15, 12, 11, 13, 25, 5, 48, 39, 47, 20, 36, 18, 44, 46, 40, 3, 23, 14, 34, 41, 16, 22, 1] | Any[22, 3, 14, 13, 47, 20, 43, 17, 27, 30, 28, 7, 36, 6, 19, 37, 18, 44, 12, 25, 5, 48, 45, 35, 4, 26, 42, 10, 24, 32, 39, 21, 33, 46, 31, 38, 16, 41, 2, 29, 34, 23, 11, 15, 40, 9, 8, 1] | (Any[8, 9, 3, 34, 23, 13, 25, 39, 10, 35, 45, 24, 32, 21, 47, 20, 36, 18, 7, 28, 6, 37, 19, 17, 43, 27, 30, 46, 15, 40, 38, 31, 44, 33, 12, 11, 14, 5, 48, 42, 26, 4, 2, 29, 41, 16, 22, 1] | (Any[9, 40, 3, 34, 41, 29, 2, 42, 26, 4, 35, 45, 10, 24, 5, 48, 13, 15, 46, 30, 43, 17, 27, 19, 37, 6, 20, 47, 21, 32, 39, 25, 14, 23, 11, 12, 33, 36, 28, 7, 18, 44, 31, 38, 8, 16, 22, 1] |

Na tej podstawie można stwierdzić, że największe znaczenie w tym wypadku ma dobór pierwszego rozwiązania.

TSP 76

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iteracje | 100 | 100 | 100 | 100 |
| Długość listy taboo | 3 | 3 | 3 | 3 |
| Miasto startowe | [60, 4, 29, 34, 31, 74, 24, 37, 20, 15, 13, 8, 30, 40, 9, 5, 26, 50, 38, 44, 64, 33, 51, 65, 47, 54, 55, 14, 27, 46, 10, 76, 61, 32, 1, 41, 45, 49, 59, 73, 70, 58, 52, 6, 68, 69, 72, 3, 39, 23, 7, 66, 12, 42, 36, 43, 62, 17, 71, 19, 28, 2, 11, 18,  25, 63, 53, 56, 16, 57, 67, 75, 22, 35, 48, 21] | [1, 41, 17, 42, 64, 32, 36, 59, 44, 33, 52, 5, 35, 26, 9, 38, 55, 71, 29, 39, 50, 14, 63, 20, 58, 51, 30, 48, 25, 66, 40, 28, 7, 75, 43, 69, 13, 3, 18, 68, 57, 74, 60, 49, 21, 31, 27, 56, 24, 34, 11, 45, 54, 22, 8, 37, 76, 72, 4, 15, 12, 67, 61, 16, 47, 73, 65, 46, 70, 10, 62, 19, 2, 23, 6, 53] | [65, 11, 23, 48, 64, 66, 5, 26, 38, 55, 58, 44, 14, 1, 59, 16, 75, 17, 20, 42, 29, 46, 10, 24, 51, 76, 57, 50, 39, 53, 45, 69, 37, 62, 25, 52, 61, 43, 6, 36, 30, 27, 13, 33, 34, 2, 32, 71, 68, 56, 35, 47, 31, 8, 40, 41, 72, 9, 49, 4, 12, 67, 7, 70, 60, 18, 74, 21, 28, 3, 19, 63, 73, 22, 54, 15] | [1, 40, 31, 33, 52, 58, 41, 13, 66, 28, 14, 63, 73, 44, 27, 22, 54, 32, 24, 50, 18, 5, 23, 11, 61, 70, 74, 4, 60, 75, 43, 69, 34, 12, 35, 19, 67, 25, 6, 57, 42, 59, 71, 53, 38, 9, 49, 56, 21, 8, 26, 55, 2, 51, 62, 20, 36, 15, 37, 29, 16, 3, 76, 7,  72, 64, 10, 46, 17, 65, 30, 45, 68, 47, 48, 39] |
| Wynik | 161986.7758824 | 180870.60203 | 156652.0878854 | 168395.280 |
| Ścieżka | Any[23, 3, 4, 20, 19, 30, 31, 12, 13, 14, 74, 15, 39, 40, 41, 60, 59, 61, 62, 64, 63, 57, 56, 65, 66, 51, 52, 27, 26, 1, 75, 76, 2, 29, 28, 58, 55, 49, 50, 73, 72, 71, 70, 67, 68, 69, 47, 5, 6, 7, 8, 9, 10, 37, 36, 38, 34, 33, 35, 18, 17, 16, 11,  32, 43, 42, 54, 53, 48, 44, 45, 46, 24, 25, 21, 22] | Any[20, 28, 43, 42, 54, 33, 34, 35, 32, 30, 31, 19, 10, 11, 16, 38, 39, 41, 40, 15, 74, 14, 13, 12, 53, 52, 49, 47, 45, 46, 26, 29, 36, 37, 18, 17, 27, 44, 48, 55, 58, 59, 60, 61, 62, 57, 56, 51, 66, 50, 24, 25, 21, 22, 23, 1, 76, 75, 2, 8, 9, 63, 64, 73, 72, 71, 65, 67, 70, 68, 69, 4, 3, 7, 6, 5] | Any[17, 36, 38, 39, 40, 55, 56, 58, 59, 60, 41, 26, 23, 1, 76, 75, 2, 8, 7, 3, 4, 22, 21, 25, 24, 46, 45, 44, 48, 47, 49, 50, 51, 52, 62, 61, 42, 43, 28, 27, 29, 32, 33, 35, 34, 54, 53, 69, 68, 70, 67, 66, 65, 57, 63, 64, 73, 72, 71, 30, 20, 5, 6, 9, 10, 12, 13, 14, 74, 15, 16, 18, 37, 31, 19, 11] | Any[40, 41, 60, 59, 52, 51, 66, 65, 57, 63, 64, 73, 72, 71, 56, 55, 33, 32, 19, 31, 30, 45, 47, 68, 67, 70, 69, 46, 23, 1, 22, 24, 25, 21, 4, 6, 12, 15, 16, 18, 38, 39, 34, 53, 50, 49, 43, 28, 29, 26, 27, 42, 58, 62, 61, 35, 36, 37, 17, 11, 10, 3, 2, 76, 75, 7, 8, 74, 14, 13, 9, 5, 20, 44, 48, 54] |

Najlepszym miastem startowym okazało się trzecie I to na nim testowane będą kolejne parametry.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iteracje | 300 | 500 | 1000 | 2000 |
| Długość listy taboo | 10 | 10 | 10 | 10 |
| wynik | 154263.42032 | 154263.42032 | 154263.42032 | 154263.42032 |
| ścieżka | Any[37, 36, 38, 39, 40, 55, 56, 58, 59, 60, 41, 30, 23, 1, 76, 75, 2, 8, 7, 3, 4, 22, 21, 25, 24, 46, 45, 44, 48, 47, 49, 50, 51, 52, 62, 61, 42, 43, 27, 26, 29, 33, 32, 35, 34, 54, 53, 69, 68, 70, 67, 66, 65, 57, 63, 64, 73, 72, 71, 28, 20, 5, 6, 9, 10, 12, 13, 14, 74, 15, 16, 11, 19, 31, 17, 18] | Any[37, 36, 38, 39, 40, 55, 56, 58, 59, 60, 41, 30, 23, 1, 76, 75, 2, 8, 7, 3, 4, 22, 21, 25, 24, 46, 45, 44, 48, 47, 49, 50, 51, 52, 62, 61, 42, 43, 27, 26, 29, 33, 32, 35, 34, 54, 53, 69, 68, 70, 67, 66, 65, 57, 63, 64, 73, 72, 71, 28, 20, 5, 6, 9, 10, 12, 13, 14, 74, 15, 16, 11, 19, 31, 17, 18] | Any[37, 36, 38, 39, 40, 55, 56, 58, 59, 60, 41, 30, 23, 1, 76, 75, 2, 8, 7, 3, 4, 22, 21, 25, 24, 46, 45, 44, 48, 47, 49, 50, 51, 52, 62, 61, 42, 43, 27, 26, 29, 33, 32, 35, 34, 54, 53, 69, 68, 70, 67, 66, 65, 57, 63, 64, 73, 72, 71, 28, 20, 5, 6, 9, 10, 12, 13, 14, 74, 15, 16, 11, 19, 31, 17, 18] | Any[37, 36, 38, 39, 40, 55, 56, 58, 59, 60, 41, 30, 23, 1, 76, 75, 2, 8, 7, 3, 4, 22, 21, 25, 24, 46, 45, 44, 48, 47, 49, 50, 51, 52, 62, 61, 42, 43, 27, 26, 29, 33, 32, 35, 34, 54, 53, 69, 68, 70, 67, 66, 65, 57, 63, 64, 73, 72, 71, 28, 20, 5, 6, 9, 10, 12, 13, 14, 74, 15, 16, 11, 19, 31, 17, 18] |

W tym przypadku długość iteracji dla listy taboo 10 nie poprawia jakości parametrów jednakże pod uwagę wezmę liczbę iteracji 1000 I dla niej przetestuję kolejne wartości długości listy taboo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iteracje | 1000 | 1000 | 1000 | 1000 |
| Długość listy taboo | 15 | 20 | 35 | 50 |
| wynik | 154405.1925 | 154228.47 | 153442.3401 | 154428.747 |
| ścieżka | Any[37, 36, 38, 39, 40, 55, 56, 58, 59, 60, 41, 30, 23, 1, 76, 75, 2, 8, 7, 3, 4, 22, 21, 25, 24, 46, 45, 44, 48, 47, 49, 50, 51, 52, 62, 61, 42, 43, 27, 26, 29, 32, 33, 35, 34, 54, 53, 69, 68, 70, 67, 66, 65, 57, 63, 64, 73, 72, 71, 28, 20, 5, 6, 9, 10, 12, 13, 14, 74, 15, 16, 11, 31, 19, 17, 18] | Any[37, 36, 38, 39, 40, 55, 56, 58, 59, 60, 41, 30, 23, 1, 76, 75, 2, 8, 7, 3, 4, 22, 21, 25, 24, 46, 45, 44, 48, 47, 49, 50, 51, 52, 62, 61, 42, 43, 27, 26, 29, 32, 33, 35, 34, 54, 53, 69, 68, 70, 67, 66, 65, 57, 63, 64, 73, 72, 71, 28, 20, 5, 6, 9, 10, 12, 13, 14, 74, 15, 16, 11, 19, 31, 17, 18] | Any[37, 36, 38, 39, 40, 55, 56, 58, 59, 60, 41, 30, 23, 1, 76, 75, 2, 8, 7, 3, 4, 22, 21, 25, 24, 46, 45, 44, 48, 50, 49, 52, 61, 62, 53, 54, 42, 43, 27, 26, 29, 32, 35, 34, 33, 28, 47, 69, 68, 70, 67, 65, 66, 57, 63, 64, 72, 73, 71, 51, 20, 5, 6, 9, 10, 12, 13, 14, 74, 15, 16, 11, 19, 31, 17, 18] | Any[37, 36, 38, 39, 40, 55, 56, 58, 59, 60, 41, 29, 23, 1, 76, 75, 2, 8, 7, 3, 4, 22, 21, 25, 24, 46, 45, 44, 48, 47, 49, 50, 51, 62, 61, 54, 42, 43, 27, 26, 28, 32, 35, 34, 33, 53, 52, 69, 68, 70, 67, 66, 65, 57, 63, 64, 73, 72, 71, 30, 20, 5, 6, 9, 10, 12, 13, 14, 74, 15, 16, 11, 19, 31, 17, 18] |

Wniosek: większa długość listy taboo poprawia wynik do pewnego stopnia.

TSP 127

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iteracje | 100 | 100 | 100 | 100 |
| Długość listy taboo | 3 | 3 | 3 | 3 |
| Miasto startowe | [105, 85, 12, 49, 99, 34, 44, 52, 45, 107, 109, 48, 123, 112, 26, 125, 60, 69, 5, 37, 102, 14, 39, 24, 90, 66, 53, 59, 98, 61, 118, 41, 117, 19, 42, 28, 79, 20, 31, 84, 83, 106, 100, 38, 72, 119, 54, 56, 63, 68, 121, 103, 35, 57,  27, 113, 4, 78, 91, 22, 71, 77, 32, 97, 21, 2, 29, 64, 114, 62, 23, 1, 124, 115, 111, 86, 92, 3, 80, 101, 6, 110, 120, 50, 46, 9, 40, 65, 126, 16, 47, 11, 7, 13, 55, 82, 18, 30, 108, 74, 70, 93, 127, 10, 116, 87, 96, 17, 88, 8, 73, 33, 67, 36, 25, 58, 94, 89, 15, 95, 51, 76, 122, 43, 104, 75, 81] | [92, 52, 51, 76, 7, 95, 112, 115, 119, 86, 45, 22, 68, 103, 38, 59, 116, 107, 44, 25, 17, 46, 34, 126, 28, 73, 13, 18, 84, 29, 82, 122, 124, 16, 89, 19, 21, 102, 123, 99, 31, 121, 43, 48, 88, 30, 66, 101, 10, 37, 27, 1, 85, 3, 60, 54, 127, 50, 47, 77, 74, 87, 104, 9, 120, 114, 56, 39, 24, 55, 63, 79, 97, 69, 4, 57, 108, 8, 49, 100, 32, 6, 26, 106, 62, 83, 40, 2, 125, 96, 93, 12, 105, 5, 20, 94, 35, 91, 64, 75, 36, 71, 53, 41, 81, 58, 78, 111, 14, 11, 70, 72, 42, 110,  61, 67, 80, 90, 33, 23, 118, 109, 15, 65, 117, 98, 113] | [44, 4, 116, 87, 95, 96, 14, 5, 127, 124, 51, 59, 85, 53, 33, 37, 89, 23, 78, 8, 81, 92, 114, 66, 40, 16, 118, 2, 19, 1, 17, 125, 110, 22, 103, 58, 21, 52, 63, 27, 117, 31, 97, 120, 86, 45, 99, 106, 71, 74, 107, 43, 18, 54, 32, 72, 67, 28, 104, 29, 70, 48, 26, 113, 10, 65, 25, 111, 24, 50, 112, 38, 36, 122, 47, 11, 64, 82, 3, 73, 84, 83, 80, 34, 60, 126, 76, 121, 109, 119, 100, 101, 115, 94, 75, 77, 62, 61, 46, 105, 15, 35, 108, 79, 6, 30, 39, 90, 88, 69, 123, 7, 12,  57, 91, 55, 49, 102, 9, 93, 98, 42, 41, 20, 56, 68, 13] | [2, 25, 106, 18, 47, 36, 16, 55, 26, 90, 63, 104, 88, 75, 85, 102, 107, 121, 83, 82, 41, 37, 11, 100, 67, 29, 76, 62, 79, 52, 70, 125, 38, 12, 54, 72, 93, 8, 39, 22, 123, 34, 7, 110, 27, 46, 28, 114, 14, 126, 80, 81, 84, 44, 68, 118, 74, 127, 32, 53, 1, 40, 3, 48, 124, 15, 86, 122, 92, 30, 60, 119, 89, 9, 42, 98, 59, 108, 91, 105, 87, 33, 65, 99, 73, 115, 24, 120, 19, 77, 4, 116, 61, 21, 56, 17, 117, 50, 95, 101, 13, 35, 78, 103, 45, 6, 49, 96, 5, 20, 112, 94, 57, 113, 69, 64, 97, 111, 43, 31, 109, 51, 58, 23, 10, 71, 66] |
| Wynik | 223779.42488 | 203420.79 | 190908.65 | 186001.1148 |
| Ścieżka | ([75, 69, 104, 92, 89, 99, 65, 93, 127, 107, 111, 112, 94, 46, 113, 125, 59, 62, 1, 37, 41, 14, 6, 24, 7, 121, 53, 49, 61, 90, 9, 72, 17, 12, 34, 43, 31, 80, 79, 126, 82, 83, 29, 25, 39, 30, 76, 119, 63, 40, 45, 103, 44, 57, 51, 2, 16, 73, 71, 70, 68, 74, 38, 42, 35, 36, 100, 64, 58, 91, 3, 105, 114, 11, 116, 60, 67, 18, 77, 78, 102, 101, 98, 54, 48, 118, 47, 56, 5, 52, 66, 55, 124, 50, 115, 106, 108, 4, 23, 21, 22, 120, 13, 10, 109, 96, 87, 88, 85, 86, 110, 8, 19, 26, 28, 95, 97, 123, 15, 20, 27, 33, 122, 32, 117, 84, 81] | [3, 52, 5, 13, 7, 6, 108, 72, 67, 86, 88, 87, 85, 110, 60, 116, 61, 90, 8, 23, 17, 27, 25, 33, 28, 122, 117, 84, 81, 126, 71, 104, 92, 99, 89, 125, 91, 24, 4, 20, 12, 14, 41, 35, 57, 121, 66, 55, 56, 37, 16, 1, 120, 115, 50, 93, 127, 107, 47, 124, 74, 68, 73, 21, 15, 114, 105, 83, 82, 102, 101, 98, 97, 123, 95, 42, 30, 106, 11, 9, 22, 31, 26, 38, 39, 34, 43, 36, 40, 44, 45, 103, 49, 118, 46, 94, 48, 53, 54, 51, 2, 10, 100, 58, 64, 113, 65, 111, 112, 62, 59, 109, 96, 119, 63, 79, 80, 32, 29, 78, 70, 69, 75, 76, 77, 18, 19] | [44, 109, 88, 87, 86, 85, 110, 71, 59, 91, 58, 64, 100, 90, 3, 120, 114, 11, 9, 60, 104, 92, 89, 125, 10, 50, 51, 2, 1, 20, 17, 21, 4, 6, 13, 115, 52, 124, 56, 121, 34, 28, 97, 98, 123, 42, 37, 16, 12, 14, 30, 43, 39, 38, 32, 29, 33, 122, 25, 26, 27, 5, 66, 113, 99, 65, 55, 127, 107, 111, 112, 94, 41, 31, 80, 79, 72, 8, 67, 73, 74, 76, 78, 117, 81, 84, 75, 96, 119, 63, 102, 101, 83, 82, 126, 77, 62, 61, 116, 105, 15, 106, 108, 23, 22, 19, 18, 69, 70, 68, 24, 7, 48, 118, 46, 47,  49, 53, 103, 93, 95, 40, 36, 35, 57, 54, 45] | [120, 114, 106, 15, 6, 108, 41, 39, 26, 81, 63, 119, 88, 86, 85, 102, 101, 98, 83, 82, 126, 79, 20, 4, 74, 75, 69, 96, 109, 87, 70, 68, 73, 72, 21, 17, 12, 14, 30, 97, 123, 95, 43, 34, 38, 32, 29, 25, 27, 31, 80, 117, 84, 78, 76, 77, 18, 22, 105, 7, 1, 16, 13, 35, 40, 42, 28, 122, 33, 9, 60, 61, 91, 100, 64, 58, 62, 59, 71, 110, 104, 92, 99, 89, 125, 11, 24, 23, 19, 8, 67, 116, 90, 3, 10, 52, 124, 112, 111, 107, 127, 93, 103, 44, 45, 94, 46, 118, 48, 49, 53, 47, 66, 113, 65, 55, 54, 57, 36, 37, 2, 51, 121, 56, 5, 50, 115] |

W tym wypadku już dla 200 iteracji został znaleziony najlepszy wynik. Poprzez małą długość listy taboo algorytm utknął w lokalnym minimum

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iteracje | 200 | 300 | 500 | 1000 |
| Długość listy taboo | 3 | 3 | 3 | 3 |
| wynik | 177202.1828 | 177202.1828 | 177202.1828 | 177202.1828 |
| ścieżka | Any[13, 120, 114, 106, 6, 108, 12, 30, 31, 79, 63, 119, 88, 86, 85, 102, 101, 98, 83, 82, 126, 76, 77, 18, 74, 75, 69, 96, 87, 109, 70, 68, 73, 72, 19, 22, 14, 41, 40, 95, 123, 97, 42, 39, 38, 25, 29, 32, 26, 27, 80, 117, 81, 84, 78, 21, 17, 20, 15, 105, 7, 1, 16, 36, 43, 34, 28, 122, 33, 4, 11, 100, 64, 58, 91, 61, 62, 59, 71, 110, 104, 99, 92, 89, 125, 60, 9, 24, 23, 8, 67, 116, 90, 3, 10, 52, 124, 112, 111, 107, 127, 93, 103, 44, 45, 94, 46, 118, 48, 53, 49, 47, 65, 113, 66,  55, 54, 57, 35, 37, 2, 51, 121, 56, 5, 50, 115] | Any[13, 120, 114, 106, 6, 108, 12, 30, 31, 79, 63, 119, 88, 86, 85, 102, 101, 98, 83, 82, 126, 76, 77, 18, 74, 75, 69, 96, 87, 109, 70, 68, 73, 72, 19, 22, 14, 41, 40, 95, 123, 97, 42, 39, 38, 25, 29, 32, 26, 27, 80, 117, 81, 84, 78, 21, 17, 20, 15, 105, 7, 1, 16, 36, 43, 34, 28, 122, 33, 4, 11, 100, 64, 58, 91, 61, 62, 59, 71, 110, 104, 99, 92, 89, 125, 60, 9, 24, 23, 8, 67, 116, 90, 3, 10, 52, 124, 112, 111, 107, 127, 93, 103, 44, 45, 94, 46, 118, 48, 53, 49, 47, 65, 113, 66,  55, 54, 57, 35, 37, 2, 51, 121, 56, 5, 50, 115] | Any[13, 120, 114, 106, 6, 108, 12, 30, 31, 79, 63, 119, 88, 86, 85, 102, 101, 98, 83, 82, 126, 76, 77, 18, 74, 75, 69, 96, 87, 109, 70, 68, 73, 72, 19, 22, 14, 41, 40, 95, 123, 97, 42, 39, 38, 25, 29, 32, 26, 27, 80, 117, 81, 84, 78, 21, 17, 20, 15, 105, 7, 1, 16, 36, 43, 34, 28, 122, 33, 4, 11, 100, 64, 58, 91, 61, 62, 59, 71, 110, 104, 99, 92, 89, 125, 60, 9, 24, 23, 8, 67, 116, 90, 3, 10, 52, 124, 112, 111, 107, 127, 93, 103, 44, 45, 94, 46, 118, 48, 53, 49, 47, 65, 113, 66,  55, 54, 57, 35, 37, 2, 51, 121, 56, 5, 50, 115] | Any[13, 120, 114, 106, 6, 108, 12, 30, 31, 79, 63, 119, 88, 86, 85, 102, 101, 98, 83, 82, 126, 76, 77, 18, 74, 75, 69, 96, 87, 109, 70, 68, 73, 72, 19, 22, 14, 41, 40, 95, 123, 97, 42, 39, 38, 25, 29, 32, 26, 27, 80, 117, 81, 84, 78, 21, 17, 20, 15, 105, 7, 1, 16, 36, 43, 34, 28, 122, 33, 4, 11, 100, 64, 58, 91, 61, 62, 59, 71, 110, 104, 99, 92, 89, 125, 60, 9, 24, 23, 8, 67, 116, 90, 3, 10, 52, 124, 112, 111, 107, 127, 93, 103, 44, 45, 94, 46, 118, 48, 53, 49, 47, 65, 113, 66,  55, 54, 57, 35, 37, 2, 51, 121, 56, 5, 50, 115] |

Zostanie teraz przetestowana długość listy taboo, dla 500 iteracji ponieważ dla mniejszej różne długości listy taboo mogą w pewnym momencie pogorszyć wyniki.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iteracje | 500 | 500 | 500 | 500 |
| Długość listy taboo | 10 | 20 | 30 | 40 |
| wynik | 176436.17397160 | 176514.772 | 168166.271 | 168170.9863 |
| ścieżka | Any[13, 120, 114, 6, 106, 15, 12, 31, 27, 79, 63, 119, 87, 86, 85, 102, 101, 98, 83, 82, 126, 76, 77, 18, 74, 75, 69, 96, 88, 109, 70, 68, 73, 72, 19, 4, 20, 41, 40, 95, 123, 97, 42, 39, 38, 33, 32, 29, 25, 26, 80, 117, 81, 84, 78, 17, 21, 22, 108, 105, 7, 1, 16, 36, 43, 34, 28, 122, 30, 14, 11, 100, 64, 58, 91, 61, 62, 59, 71, 110, 104, 99, 92, 89, 125, 60, 9, 24, 23, 8, 67, 116, 90, 3, 10, 52, 124, 112, 111, 107, 127, 93, 103, 44, 45, 94, 46, 118, 48, 53, 49, 47, 65, 113, 66,  55, 54, 57, 35, 37, 2, 51, 121, 56, 5, 50, 115] | Any[13, 120, 114, 6, 106, 15, 12, 31, 27, 79, 63, 119, 88, 86, 85, 102, 101, 98, 83, 82, 126, 76, 77, 18, 74, 75, 69, 96, 87, 109, 70, 68, 73, 72, 19, 22, 4, 36, 40, 95, 123, 97, 42, 39, 38, 33, 29, 32, 25, 26, 80, 117, 81, 84, 78, 21, 17, 20, 108, 105, 7, 1, 16, 41, 43, 34, 28, 122, 30, 14, 11, 100, 64, 58, 91, 61, 62, 59, 71, 110, 104, 99, 92, 89, 125, 60, 9, 24, 23, 8, 67, 116, 90, 3, 10, 52, 124, 112, 111, 107, 127, 93, 103, 44, 45, 94, 46, 118, 48, 53, 49, 47, 65, 113, 66  55, 54, 57, 35, 37, 2, 51, 121, 56, 5, 50, 115] | Any[115, 120, 114, 6, 106, 15, 12, 27, 31, 79, 85, 86, 88, 119, 63, 102, 101, 98, 83, 82, 126, 76, 77, 18, 74, 75, 69, 96, 87, 109, 70, 68, 73, 72, 19, 22, 20, 41, 40, 95, 123, 97, 42, 39, 28, 122, 29, 32, 33, 25, 80, 117, 81, 84, 78, 21, 17, 4, 108, 105, 7, 1, 16, 36, 43, 34, 38, 26, 30, 14, 11, 100, 64, 58, 91, 61, 62, 59, 71, 110, 104, 99, 92, 89, 125, 60, 9, 24, 23, 8, 67, 116, 90, 3, 10, 52, 124, 112, 111, 107, 127, 93, 103, 44, 45, 94, 46, 118, 48, 53, 49, 47, 65, 113, 66, 55, 54, 57, 35, 37, 2, 51, 121, 56, 5, 50, 13] | Any[13, 120, 114, 6, 108, 20, 12, 31, 27, 79, 85, 86, 88, 119, 63, 102, 101, 98, 83, 82, 126, 76, 77, 18, 74, 75, 69, 96, 87, 109, 70, 68, 73, 72, 19, 4, 14, 41, 40, 95, 123, 97, 42, 39, 28, 122, 29, 32, 33, 25, 80, 117, 81, 84, 78, 21, 17,  22, 106, 105, 7, 1, 16, 36, 43, 34, 38, 26, 30, 15, 11, 100, 64, 58, 91, 61, 62, 59, 71, 110, 104, 99, 92, 89, 125, 60, 9, 24, 23, 8, 67, 116, 90, 3, 10, 52, 124, 112, 111, 107, 127, 93, 103, 44, 45, 94, 46, 118, 48, 53, 49, 47, 65, 113, 66,  55, 54, 57, 35, 37, 2, 51, 121, 56, 5, 50, 115] |

Jak można zauważyć do długości taboo 30 wartości funkcji celu poprawiają się.

Wnioski:

W algorytmie taboo search istnieje w zależności od liczby iteracji pewna długość listy taboo która poprawia wyniki jednakże najważniejsze jest aby dobrać dobre początkowe rozwiązanie.