Zadanie 2. Lokalne przeszukiwanie

Oskar Kiliańczyk 151863 & Wojciech Kot 151876

1 Opis zadania

Zadanie polega na implementacji lokalnego przeszukiwania w wersjach stromej (steepest) i zachłannej (greedy), z dwoma różnym rodzajami sąsiedztwa, startując albo z rozwiązań losowych, albo z rozwiązań uzyskanych za pomocą jednej z heurystyk opracowanych w ramach poprzedniego zadania. W sumie 8 kombinacji — wersji lokalnego przeszukiwania. Jako punkt odniesienia należy zaimplementować algorytm losowego błądzenia, który w każdej iteracji wykonuje losowo wybrany ruch (niezależnie od jego oceny) i zwraca najlepsze znalezione w ten sposób rozwiązanie. Algorytm ten powinien działać w takim samym czasie jak średnio najwolniejsza z wersji lokalnego przeszukiwania.

1.1 Sąsiedztwa

W przypadku rozważanego problemu potrzebne będą dwa typy ruchów:

- ruchy zmieniające zbiory wierzchołków tworzące dwa cykle,
- ruchy wewnątrztrasowe, które jedynie zmieniają kolejność wierzchołków na trasie.

Stosujemy dwa rodzaje ruchów wewnątrztrasowych (jeden albo drugi, stąd dwa rodzaje sąsiedztwa). jeden to wymiana dwóch wierzchołków wchodzących w skład trasy, drugi to wymiana dwóch krawędzi.

1.2 Randomizacja kolejności przeglądania dla algorytmów zachłannych

W obu wersjach algorytmów zachłannych stosujemy randomizację wyboru. Dla ruchów wewnątrz cykli wykonujemy lokalne zamiany elementów w obrębie jednej ścieżki. Randomizacja polega na losowym przetasowaniu indeksów w obrębie dostępnych miast, co zmienia kolejność, w jakiej będą przetwarzane. W przypadku algorytmu wykorzystującego zmiany pomiędzy ścieżkami zmiana odbywa się pomiędzy dwoma cyklami. Tworzymy listę możliwych par punktów do zamiany między dwoma cyklami. Losowo przetasowujemy te możliwe pary, dzięki czemu kolejność przetwarzania nie jest deterministyczna. Losowo wybieramy jedną parę do wymiany (w której obliczenie zmiany w funkcji celu daje poprawę) i przeprowadzamy zamianę.

Instance	Algorithm	Start Alg.	Best | Avg | Worst	BestT | AvgT | WorstT	BestD | AvgD
kroA200	greedy E	random	44904 | 47506.6 | 49675	0.047 | 0.001 | 0.127	325030 | 292617
kroA200	greedy E	własny	30310 | 33201.2 | 37119	0.004 | 0.000 | 0.021	322 | 96.77
kroA200	greedy V	random	96986 | 120985 | 165848	0.018 | 0.000 | 0.100	250459 | 219127
kroA200	greedy V	własny	30435 | 32934.3 | 36065	0.004 | 0.000 | 0.062	2099 | 149.55
kroA200	steepest E	random	45216 | 49743.6 | 54231	0.401 | 0.006 | 0.990	321512 | 289999
kroA200	steepest E	własny	30293 | 32925.6 | 36894	0.004 | 0.000 | 0.039	1028 | 94.99
kroA200	steepest V	random	101839 | 122314 | 144402	0.274 | 0.004 | 1.593	247067 | 215586
kroA200	steepest V	własny	30590 | 32628.5 | 36593	0.002 | 0.000 | 0.033	1912 | 150.34
kroA200	steepest E+V	random	38834 | 43060 | 47763	1.476 | 0.017 | 2.796	334276 | 297054
kroA200	steepest E+V	własny	30310 | 32554.1 | 36373	0.007 | 0.001 | 0.158	3050 | 363.02
kroA200	random	random	306319 | 335290 | 376971	2.796 | 0.028 | 2.797	37983 | 7827.03
kroA200	random	własny	30426 | 32936.3 | 37192	2.797 | 0.028 | 2.855	0 | 0
kroB200	greedy E	random	44426 | 47039.6 | 49813	0.053 | 0.001 | 0.116	315327 | 285202
kroB200	greedy E	własny	30863 | 33215.9 | 35995	0.005 | 0.000 | 0.018	1247 | 213.43
kroB200	greedy V	random	99971 | 118645 | 150580	0.032 | 0.001 | 0.143	244988 | 213139
kroB200	greedy V	własny	31218 | 33223.5 | 36519	0.004 | 0.000 | 0.027	1339 | 122.56
kroB200	steepest E	random	45648 | 49405.5 | 53560	0.427 | 0.005 | 0.898	304743 | 283819
kroB200	steepest E	własny	30934 | 33242 | 36790	0.004 | 0.000 | 0.036	1189 | 246.88
kroB200	steepest V	random	96970 | 118903 | 142320	0.297 | 0.005 | 0.863	252946 | 212667
kroB200	steepest V	własny	31341 | 33171.5 | 36555	0.002 | 0.000 | 0.038	1339 | 214.25
kroB200	steepest E+V	random	38739 | 43252.6 | 47376	1.633 | 0.024 | 3.000	312702 | 289492
kroB200	steepest E+V	własny	31156 | 33014.7 | 36479	0.007 | 0.000 | 0.194	4046 | 496.52
kroB200	random	random	298060 | 324327 | 346475	3.000 | 0.030 | 3.028	32259 | 8276.57
kroB200	random	własny	31466 | 33448.2 | 36604	3.028 | 0.030 | 3.059	0 | 0

Tabela 1: Wyniki dla różnych instancji i algorytmów

2 Opisy algorytmów

2.1 Zmiany wewnątrz cyklu

2.1.1 Zachłanny

2.1.2 Steepest

2.2 Zmiany pomiędzy cyklami

2.2.1 Zachłanny

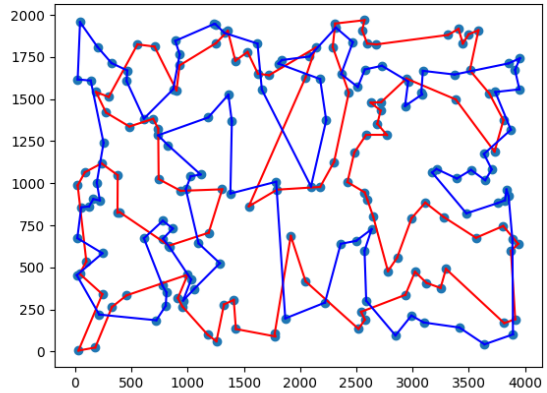
2.2.2 Steepest

3 Wyniki

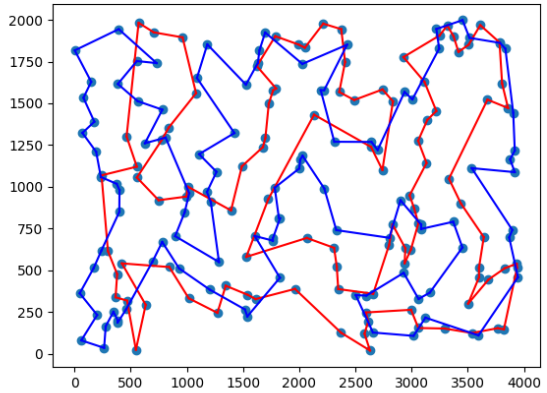
3.1 Tabela wynikowa

3.2 Wizualizacja wyników

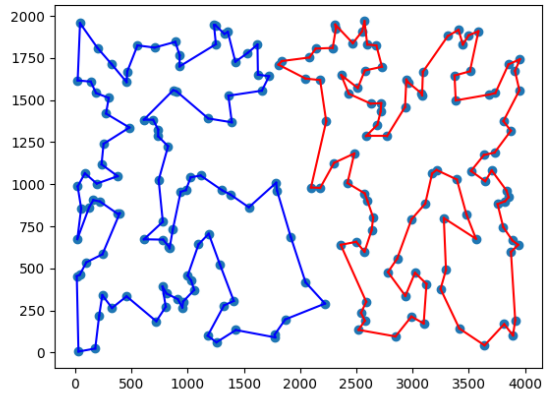
3.2.1 Algorytm wymiany krawędzi wewnątrz cykli (zachłanny)



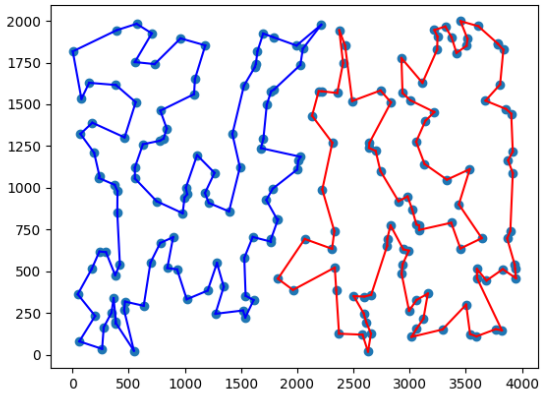
Rysunek 1: kroA200, losowy start



Rysunek 2: kroB200, losowy start

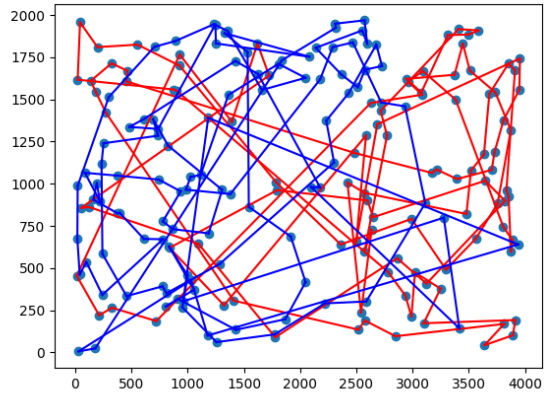


Rysunek 3: kroA200, własny algorytm startowy

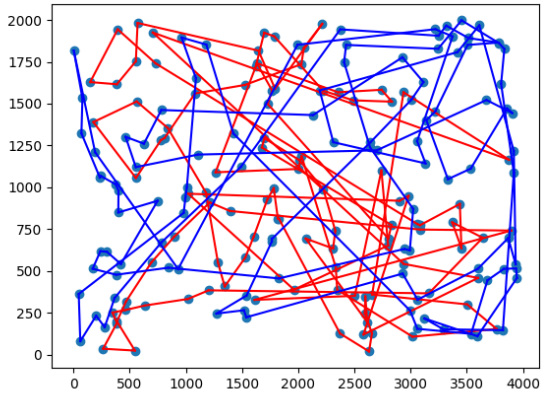


Rysunek 4: kroB200, własny algorytm startowy

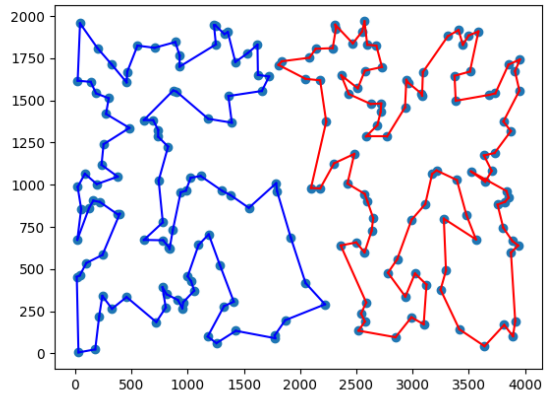
3.2.2 Algorytm wymiany wierzchołków między cyklami (zachłanny)



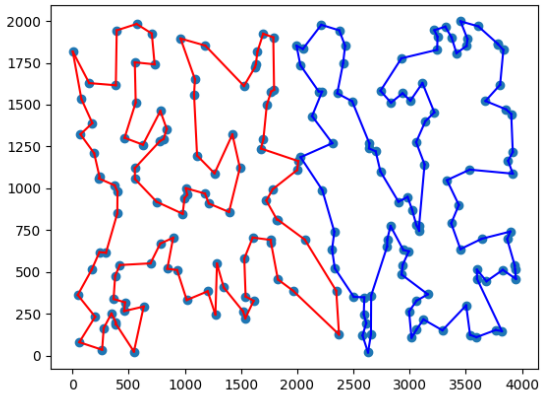
Rysunek 5: kroA200, losowy start



Rysunek 6: kroB200, losowy start

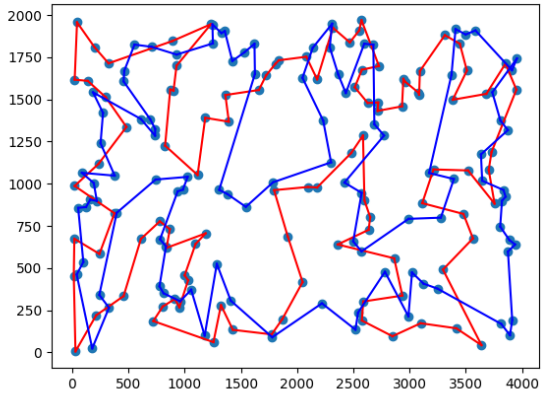


Rysunek 7: kroA200, własny algorytm startowy

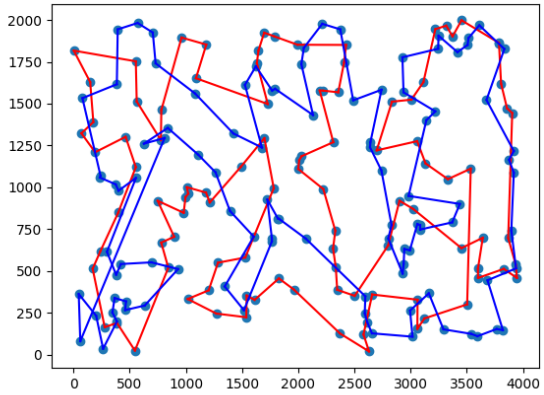


Rysunek 8: kroB200, własny algorytm startowy

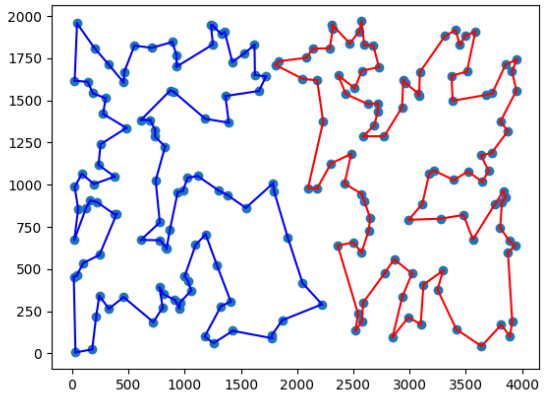
3.2.3 Algorytm wymiany krawędzi wewnątrz cykli (steepest)



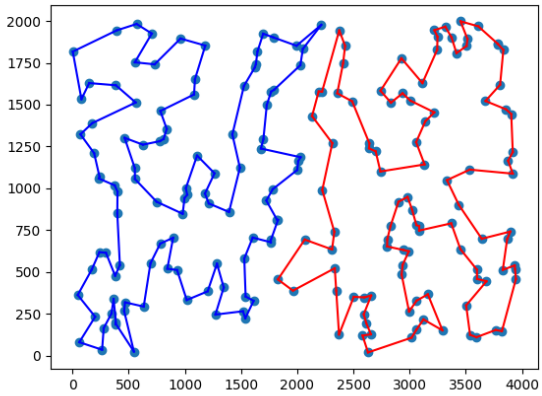
Rysunek 9: kroA200, losowy start



Rysunek 10: kroB200, losowy start

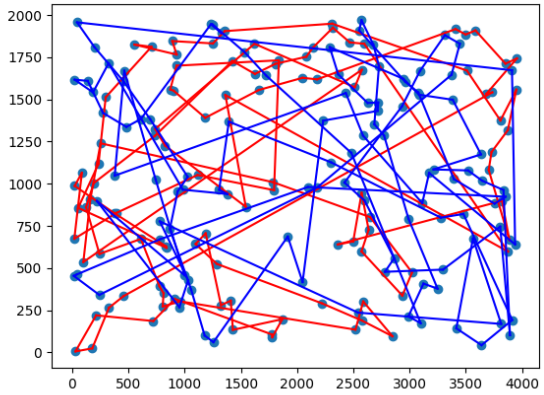


Rysunek 11: kroA200, własny algorytm startowy

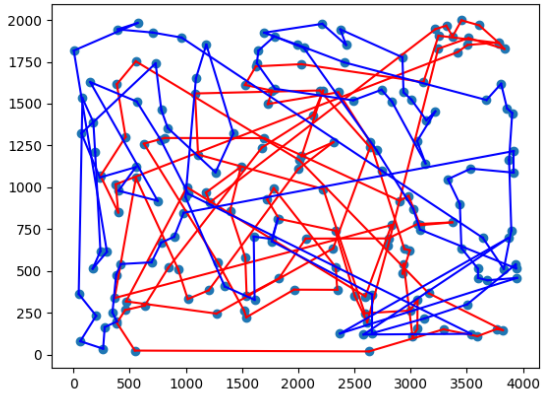


Rysunek 12: kroB200, własny algorytm startowy

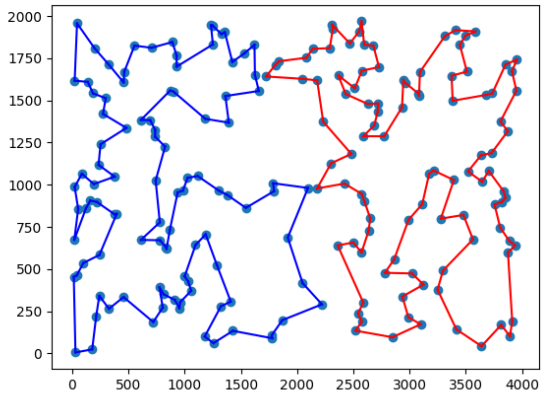
3.2.4 Algorytm wymiany wierzchołków między cyklami (steepest)



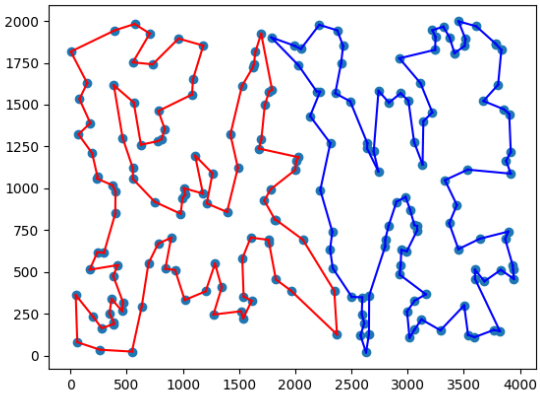
Rysunek 13: kroA200, losowy start



Rysunek 14: kroB200, losowy start

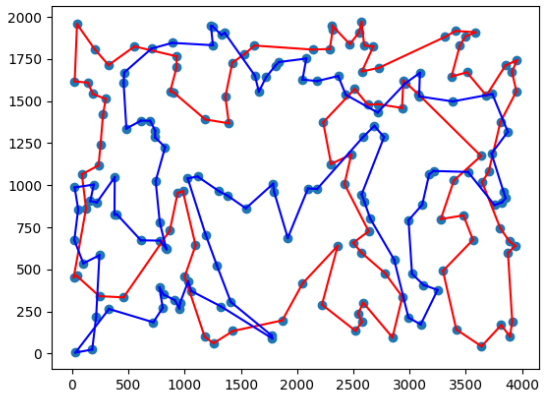


Rysunek 15: kroA200, własny algorytm startowy

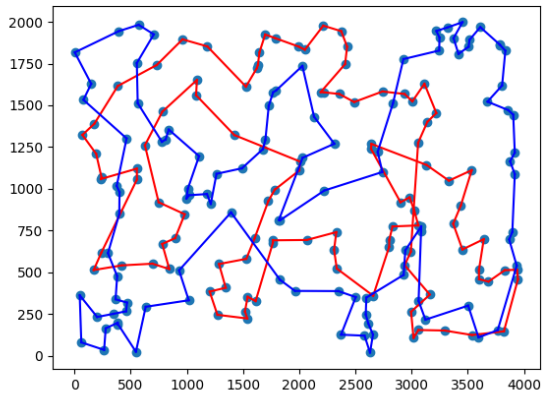


Rysunek 16: kroB200, własny algorytm startowy

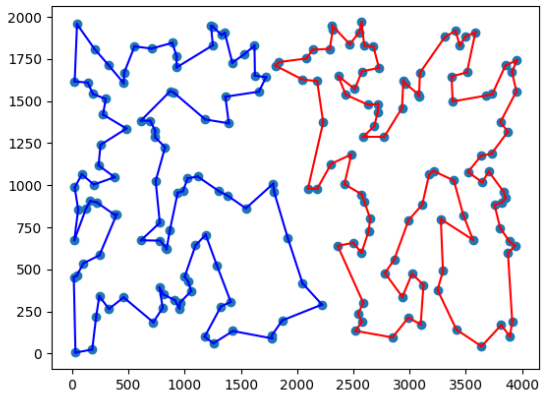
3.2.5 Algorytm steepest dla obu typów sąsiedztwa



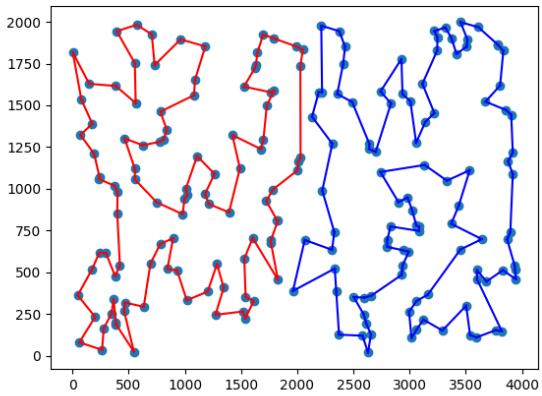
Rysunek 17: kroA200, losowy start



Rysunek 18: kroB200, losowy start

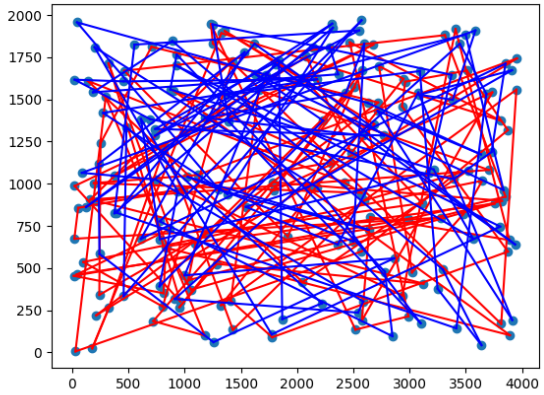


Rysunek 19: kroA200, własny algorytm startowy

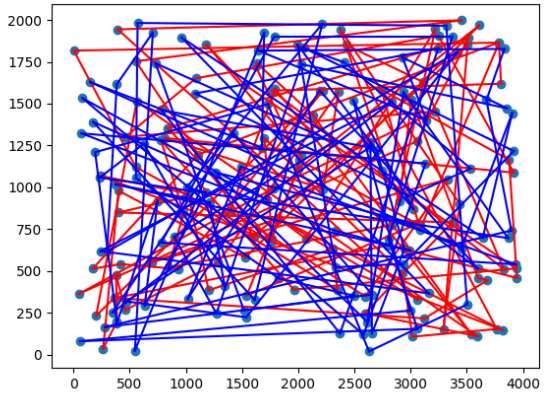


Rysunek 20: kroB200, własny algorytm startowy

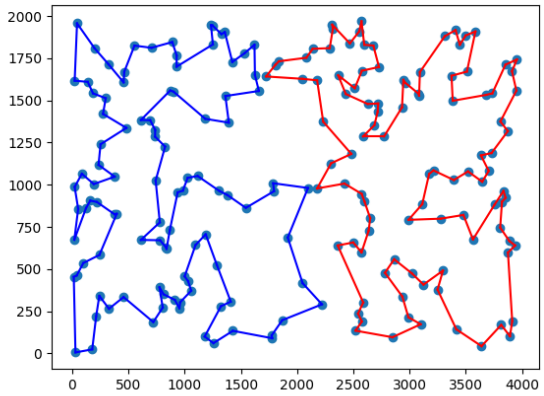
3.2.6 Algorytm losowych ruchów w obu typach sąsiedztwa (random)



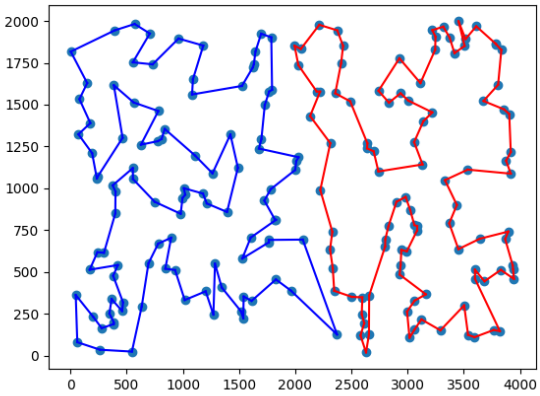
Rysunek 21: kroA200, losowy start



Rysunek 22: kroB200, losowy start



Rysunek 23: kroA200, własny algorytm startowy



Rysunek 24: kroB200, własny algorytm startowy

4 Link do repozytorium

Kod źródłowy w repozytorium GitHub dostępny pod linkiem:
Repozytorium Local Search.