

Sprawozdanie

Algorytmy Metaheurystyczne - Laboratorium

Radosław Wojtczak, Witold Karaś

1 Wstęp:

1.1 Algorytm k-random

Algorytm k-random polega na wygenerowaniu losowego dopuszczalnego rozwiązania (co w naszym przypadku, gdzie wszystkie grafy są gęste, oznacza wybranie losowej permutacji wierzchołków) i wyznaczenie dla niego funkcji celu. Ów schemat powtarzamy k razy, z czego wybieramy najmniejszą drogę co do wartości.

Złożoność obliczeniowa: Procedura *k-random* wykonuje n razy funkcję *cost*, która działa w czasie liniowym. Dodatkowo do generowania nowych rozwiązań korzystamy z wbudowanej biblioteki pythona o nazwie *random*, która zawiera funkcję *shuffle*, której złożoność obliczeniowa wynosi $O(n)$, gdyż korzysta z "Fisher–Yates-shuffle". Ostatecznie więc, złożoność obliczeniowa tej metody wynosi $O(k * n)$.

Złożoność pamięciowa: Procedura *k-random* generuje n-elementową permutację wierzchołków w formie listy. Następnie w specjalnej zmiennej przechowywana jest lista o najmniejszej funkcji celu, stąd ostatecznie nasza funkcja składa się z dwóch list o długości n. Wniosek: $O(n)$

1.2 Algorytm najbliższego sąsiada: omówienie

Algorytm najbliższego sąsiada jest algorytmem z grupy algorytmów zachłannych- wybiera lokalnie najlepsze rozwiązanie, w ten sposób budując pełne rozwiązanie.

Złożoność obliczeniowa: Procedura *nearestneighbor* wykonuje się aż do momentu, gdy cała lista *path* zostanie zapełniona. W każdym kroku dodajemy do niej jeden wierzchołek, więc zewnętrzny while wykonuje się dokładnie n razy. Następnie wewnątrz while'a przechodzimy liniowo po strukturze w poszukiwaniu sąsiada, do której jeszcze nie dotarliśmy (złożoność liniowa), jednakże dodatkowo dokonuje sortowania wierzchołków w celu ułatwienia iterowania po sąsiadach. Do sortowania korzystamy z wbudowanej funkcji *sorted*, która wykonuje się w czasie $O(n * \log n)$. Ostatecznie otrzymujemy złożoność obliczeniową worst-case na poziomie $O(n^2 * \log n)$.

Złożoność pamięciowa: Wewnątrz algorytmu dokonujemy tworzenia dodatkowego słownika, stąd wnioskujemy, iż złożoność pamięciowa tego algorytmu jest równa $O(n)$.

1.3 Algorytm rozszerzonego najbliższego sąsiada: omówienie

Algorytm rozszerzonego najbliższego sąsiada polega na odizolowaniu działania algorytmu od wyboru wierzchołka początkowego poprzez wykonanie podstawowego sąsiada dla wszystkich wierzchołków startowych **Złożoność:** Algorytm wykorzystuje wszystkie funkcje algorytmu najbliższego sąsiada z tą różnicą, iż iteruje po wszystkich wierzchołkach, których jest dokładnie n , dlatego złożoność obliczeniowa tego algorytmu jest równa $O(n^3 * \log n)$. Złożoność pamięciowa nie ulega zmianom.

1.4 Algorytm 2-opt

Algorytm 2-opt jest algorytmem, który korzysta z własności przestrzeni euklidesowych zwanej nierównością trójkątą. Zauważa, iż jeśli drogi są ze sobą "splątane" to ich rozwinięcie spowoduje zmniejszenie funkcji celu. Na tej podstawie posiadając permutację wejściową dokonujemy inwersji pewnej części tablicy w oczekiwaniu, iż doszło do takowych rozplątania. **Złożoność obliczeniowa:** Ze względu na występującą pętlę while, która wykonuje się do momentu, aż nie natrafimy na ulepszenie aktualnej permutacji bardzo trudno określić złożoność obliczeniową tego algorytmu. Na podstawie testów dla przestrzeni Euklidesowej wnioskujemy, iż ów algorytm może pracować nawet w złożoności $O(n^3)$. Sytuacja komplikuje się, gdy przechodzimy w przestrzenie, które nie spełniają nierówności trójkątą. Intuicja może nam podpowiadać, iż w tej sytuacji możemy osiągnąć aż wykładniczą złożoność w najgorszym przypadku, jednakże żadne z naszych testów nie potwierdziły tej tezy. **Złożoność pamięciowa:** Ze względu na to, iż w procedurze tworzymy jedną listę n elementową przechowującą aktualną permutację, i jedną listę n elementową przechowującą najlepszą permutację, złożoność pamięciowa sprowadza się do $O(n)$.

2 Przypadek Testowy 1- Najbliższy sąsiad, a rozszerzony najbliższy sąsiad- porównanie wyników oraz czasów wykonywania

2.1 Cel:

W tej części zostaną ze sobą porównane dwie bardzo podobne metody rozwiązywania problemu komiwojażera- algorytm najbliższego sąsiada oraz rozszerzony algorytm najbliższego sąsiada. Ze względu na to, iż w rozszerzonym sąsiedzie aby odizolować otrzymany wynik od wierzchołka początkowego, przechodzimy po wszystkich wierzchołkach, naturalnym jest, iż spodziewamy się otrzymywać lepszy wynik w nieco dłuższym czasie. Celem tego testu będzie zbadanie, czy w każdym przypadku stosowanie rozszerzonego algorytmu najbliższego sąsiada jest zdecydowanie lepsze od swojego poprzednika.

2.2 Założenia:

Do tego badania użyto automatycznie wygenerowanych grafów typu **EUC2D**. Rozmiary grafów (oznaczane literą n) należą do zbioru $n \in \{10, 15, 20, \dots, 100\}$. Dla algorytmu najbliższego sąsiada dla zadanej długości grafu badanie powtórzone 5 razy.

2.3 Wyniki:

Poniższe tabele przedstawiają wyniki wykonanych testów:

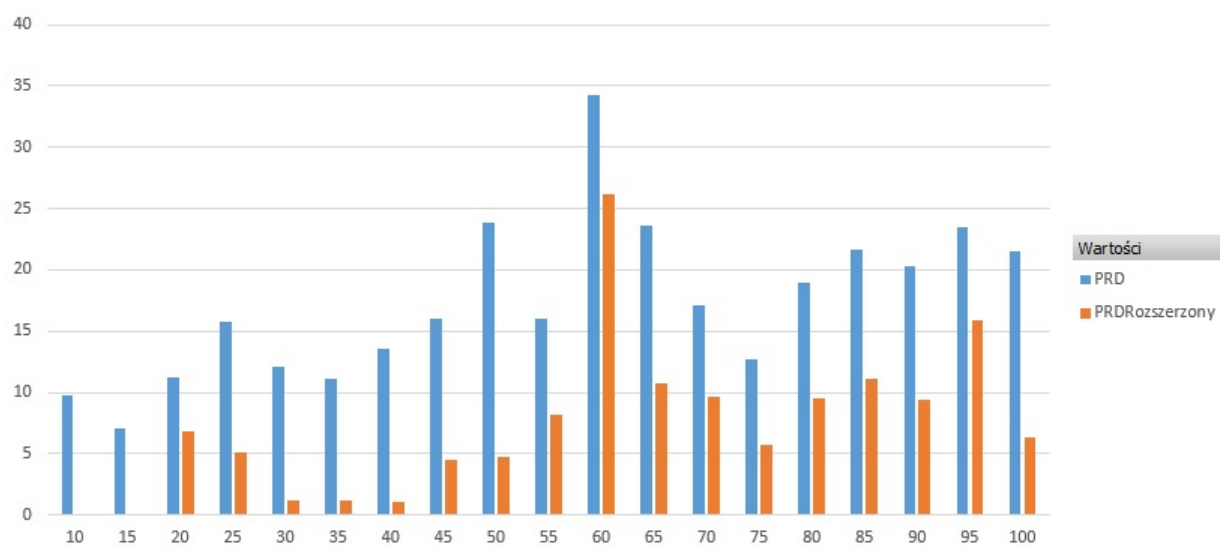
n	Najbliższy	Rozszerzony
10	9,786950732	0
15	7,120743034	0
20	11,19483315	6,779661017
25	15,79634465	5,147058824
30	12,10191083	1,193317422
35	11,13360324	1,126126126
40	13,58260206	1,091703057
45	16,03927987	4,469273743
50	23,79562044	4,744525547
55	16,00127755	8,202443281
60	34,19556566	26,15629984
65	23,64657814	10,74626866
70	17,08344301	9,612625538
75	12,67305644	5,747126437
80	18,91247317	9,574468085
85	21,60317111	11,11111111
90	20,33256157	9,341317365
95	23,46057932	15,91148577
100	21,46546776	6,385404789

Tabela 1: Współczynnik PRD dla otrzymanych wyników. Za wartość referencyjną posłużył wynik otrzymany przez algorytm 2-opt z elementem startowym wyznaczonym przez najbliższego sąsiada

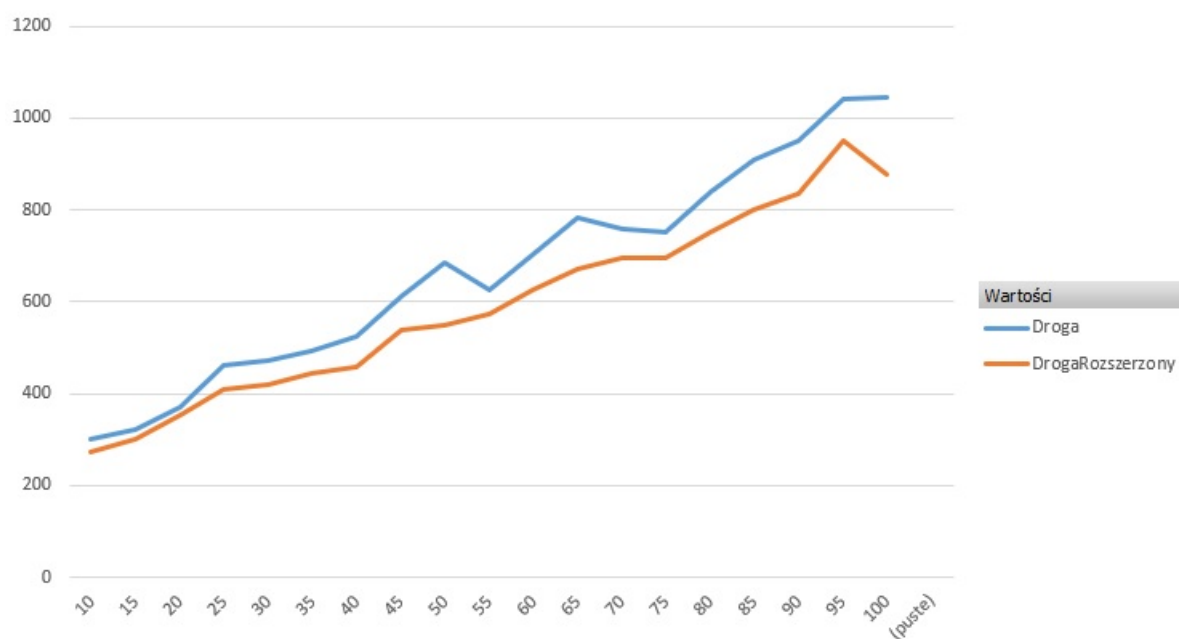
n	DrogaNajbliższy	CzasNajbliższy	DrogaRozszerzony	CzasRozszerzony
10	300,4	0,192	271	1,97
15	323	0,41	300	1,95
20	371,6	0,646	354	125,88
25	459,6	0,996	408	23,74
30	471	1,4	419	39,84
35	494	1,838	444	64,69
40	524,2	2,372	458	98,93
45	611	3,004	537	145,38
50	685	3,998	548	180,72
55	626,2	4,782	573	249,38
60	703,6	5,494	627	313,86
65	783,2	6,778	670	394,06
70	759,8	7,352	697	496,95
75	751,2	8,576	696	496,95
80	838,6	9,606	752	611,32
85	908,2	11,43	801	864,69
90	950,2	11,932	835	1042,41
95	1042,6	13,768	949	1244,87
100	1045,4	14,204	877	1430,27

Tabela 2: Wyniki dróg i czasów dla rozpatrywanych wariantów (należy pamiętać, że dla algorytmu najbliższego sąsiada są to wartości **uśrednione!**. Jednostką czasu są milisekundy)

2.4 Wykresy:



Rysunek 1: Współczynnik PRD dla testowanych algorytmów



Rysunek 2: Osiągnięte minimalne funkcje celu dla testowanych algorytmów

2.5 Wnioski:

Zauważamy, iż mimo otrzymywania lepszych wyników dla rozszerzonego algorytmu najbliższego sąsiada, czasy, w jakich je otrzymujemy są zdecydowanie gorsze niż dla najbliższego sąsiada dla losowo wybranego elementu początkowego. Dodatkowo różnice w otrzymanych drogach są niewielkie, co powoduje, iż nie zawsze korzystanie z algorytmu rozszerzonego sąsiada jest optymalne. Należy dodatkowo zauważyć, iż na różnice czasowe mogą wpływać takie aspekty jak: sprzęt, na którym wykonywano testy oraz język programowania, w którym ów test został wykonany.

3 przypadek testowy 2

3.1 Cel:

Celem badania jest sprawdzenie czy rzeczywista złożoność czasowa dla algorytmu k-random jest zgodna z oczekiwanym $O(n)$.

3.2 Założenia:

Podczas badania korzystać będziemy z losowych instancji grafów pełnych, symetrycznych o $n \in \{100, 200, 300 \dots 1000\}$ wierzchołkach oraz ustalonym $k = 1000$. Badanie czasu każdego rozmiaru grafów zostało powtórzone 100 razy.

3.3 Wyniki:

Tabela czasów: przedstawia czasy wykonania programu. Dane w tabach, dla czytelności, zostały zaokrąglone do 3 miejsc po przecinku. W dalszych obliczeniach korzystaliśmy z danych dokładnych zamieszczonych w pliku csv/xlsx.

n	100	200	300	400	500	600	700	800	900	1000
T [s]	0.781	0.810	0.860	0.903	0.922	0.957	0.996	1.007	1.044	1.074
SD	0.009	0.010	0.223	0.033	0.037	0.028	0.029	0.020	0.020	0.012
SE	0.001	0.001	0.022	0.003	0.004	0.003	0.003	0.002	0.002	0.001

Tabela 3: T - średni czas wykonania (w sekundach), SD - odchylenie standardowe, SE - błąd standardowy

Odchylenie standardowe oraz błąd standardowy zostały obliczone według wzorów:

$$\sigma = \sqrt{\frac{\sum_{n=1}^{100} (\bar{x} - x_n)^2}{100}} \quad (1)$$

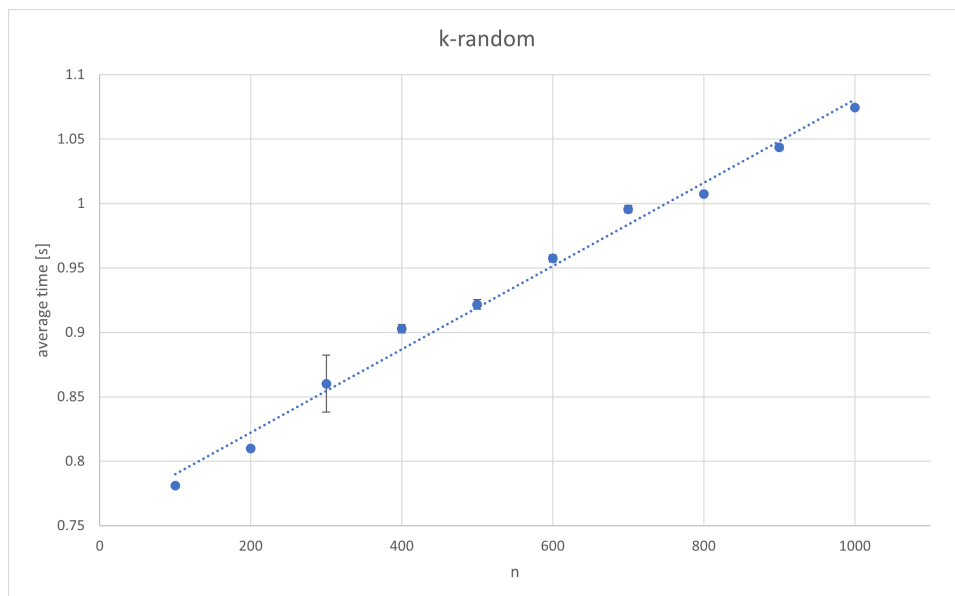
odchylenie standardowe 1

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{100}} \quad (2)$$

błąd standardowy 2

3.4 Wykresy:

Na osi OX wykresu naniesione zostały n dla których wykonywany był pomiar, na osi OY uśrednione wartości czasów wykonania wraz z zaznaczonymi błędami standardowymi. Zauważyć można że punkty wskazujące średnie czasy wykonania ustaiwione są w zależności liniowej.



Rysunek 3: średnie czasy wykonania wraz z błędami standardowymi

3.5 Wnioski:

Zmierzone dane potwierdziły tezę, że złożoność czasowa metody k-random przy ustalonym k jest liniowa względem wielkości grafu.

3.6 Tabela czasów:

100	200	300	400	500	600	700	800	900	1000
0.791	0.803	0.855	0.975	0.895	0.921	1.063	1.008	1.028	1.073
0.776	0.819	0.831	0.961	0.937	0.921	1.020	1.006	1.032	1.081
0.790	0.810	0.836	0.934	0.963	0.946	1.026	1.019	1.044	1.076
0.783	0.817	0.836	0.942	0.945	0.971	1.038	1.000	1.033	1.080
0.788	0.808	0.834	0.932	0.935	0.917	1.022	1.025	1.039	1.062
0.783	0.803	0.841	0.929	0.938	0.942	1.022	1.011	1.032	1.090
0.784	0.815	0.843	0.926	0.898	0.930	1.042	1.007	1.034	1.105
0.780	0.806	0.829	0.968	0.892	0.921	1.025	1.006	1.036	1.099
0.781	0.813	0.853	0.955	0.918	0.909	1.033	0.999	1.016	1.082
0.786	0.800	0.838	0.936	0.918	0.922	1.030	1.003	1.039	1.091
0.778	0.805	0.850	0.912	0.953	0.931	1.028	1.009	1.036	1.075
0.786	0.811	0.833	0.954	0.961	0.922	1.027	1.004	1.031	1.091
0.772	0.812	0.836	0.866	0.898	0.925	1.027	0.994	1.054	1.079
0.784	0.810	0.834	0.856	0.886	0.941	1.035	0.993	1.027	1.080
0.784	0.812	0.846	0.862	0.917	0.943	1.032	1.000	1.023	1.095
0.785	0.803	0.833	0.873	0.949	0.936	1.022	1.000	1.030	1.071
0.793	0.812	0.840	0.862	1.006	0.946	1.063	1.019	1.021	1.093
0.789	0.797	0.825	0.868	1.000	0.922	1.070	0.993	1.034	1.084
0.781	0.807	0.839	0.859	1.034	0.918	1.038	0.996	1.039	1.061
0.793	0.856	0.842	0.872	1.051	0.955	1.028	1.021	1.032	1.094
0.768	0.799	0.837	0.868	1.032	0.986	1.034	1.010	1.039	1.097
0.790	0.839	0.834	0.871	0.974	0.944	1.040	1.012	1.036	1.078
0.785	0.810	0.833	0.863	0.921	1.006	1.033	1.007	1.047	1.105
0.775	0.815	0.841	0.863	0.927	1.010	1.031	0.993	1.034	1.066
0.786	0.800	0.840	0.863	0.921	0.939	1.035	0.994	1.030	1.079
0.781	0.821	0.842	0.892	0.913	0.948	1.019	1.016	1.025	1.078
0.785	0.814	0.829	0.935	0.974	0.944	0.971	1.026	1.026	1.079
0.797	0.812	0.816	0.915	0.988	0.972	0.975	1.033	1.040	1.078
0.790	0.799	0.843	0.932	0.947	0.949	0.982	1.067	1.027	1.066
0.775	0.861	0.831	0.876	0.933	0.998	0.969	1.020	1.026	1.075
0.780	0.792	0.833	0.872	0.937	0.950	0.958	1.005	1.025	1.073
0.776	0.812	0.827	0.887	0.908	0.950	0.971	0.986	1.032	1.075
0.788	0.813	0.831	0.870	0.964	0.933	0.969	0.976	1.040	1.094
0.785	0.797	0.832	0.916	0.976	0.902	0.962	1.035	1.034	1.073
0.783	0.814	0.840	0.900	0.944	0.945	0.979	1.004	1.040	1.078
0.768	0.814	0.829	0.906	0.915	0.966	0.964	1.010	1.034	1.070
0.769	0.812	0.831	0.896	0.960	0.911	0.980	1.041	1.066	1.066
0.789	0.808	0.836	0.910	0.956	0.932	0.974	1.009	1.050	1.074
0.779	0.807	0.831	0.905	0.977	1.009	0.964	0.977	1.031	1.073
0.783	0.799	0.829	0.933	0.912	0.993	0.982	1.010	1.029	1.074
0.780	0.808	0.833	0.918	0.943	0.975	0.971	1.004	1.033	1.062
0.767	0.802	0.830	0.941	0.925	0.994	0.973	0.982	1.028	1.080
0.793	0.815	0.841	0.912	0.957	1.005	0.968	1.039	1.090	1.092
0.786	0.807	0.832	0.927	0.923	0.973	0.970	0.990	1.045	1.075
0.778	0.815	0.832	0.938	0.928	1.012	0.986	0.988	1.051	1.064
0.767	0.812	0.833	0.937	0.943	1.014	0.981	0.980	1.038	1.086
0.788	0.815	0.827	0.925	0.956	1.027	0.960	1.041	1.035	1.069
0.780	0.815	0.838	0.871	0.927	1.000	0.963	1.000	1.022	1.083
0.778	0.818	0.844	0.869	0.971	1.004	0.983	1.044	1.022	1.070
0.788	0.806	0.830	0.910	0.943	0.998	0.966	0.965	1.019	1.084

100	200	300	400	500	600	700	800	900	1000
0.776	0.806	0.833	0.902	0.939	1.008	1.017	1.020	1.018	1.062
0.773	0.808	0.839	0.885	0.911	0.998	0.972	0.974	1.037	1.068
0.794	0.803	0.837	0.885	0.895	0.989	0.969	0.996	1.041	1.080
0.775	0.817	0.844	0.890	0.917	0.949	0.967	0.981	1.029	1.067
0.774	0.812	0.841	0.901	0.938	0.963	0.970	0.994	1.022	1.077
0.766	0.806	0.835	0.905	0.920	0.945	0.972	0.977	1.025	1.072
0.796	0.807	0.826	0.929	0.896	0.945	0.976	0.968	1.067	1.077
0.774	0.791	0.830	0.893	0.903	1.003	0.980	0.975	1.066	1.071
0.793	0.821	0.828	0.914	0.871	1.015	0.976	0.967	1.029	1.061
0.765	0.799	0.824	0.952	0.892	0.999	0.969	1.007	1.020	1.088
0.791	0.799	0.837	0.956	0.867	0.973	0.982	1.015	1.047	1.069
0.778	0.813	0.843	0.953	0.878	0.947	0.965	0.985	1.050	1.060
0.782	0.808	0.841	0.948	0.887	0.965	0.966	1.010	1.026	1.057
0.774	0.797	0.841	0.947	0.872	0.952	0.970	1.031	1.026	1.069
0.781	0.819	0.839	0.894	0.877	0.960	0.976	0.970	1.094	1.059
0.784	0.802	0.836	0.869	0.894	0.948	0.975	1.029	1.047	1.083
0.782	0.818	0.849	0.878	0.908	0.956	0.970	1.053	1.044	1.071
0.781	0.811	0.843	0.877	0.900	0.961	0.963	1.049	1.028	1.061
0.805	0.821	0.843	0.861	0.926	0.954	0.969	1.056	1.042	1.061
0.776	0.803	0.828	0.871	0.902	0.974	0.980	1.013	1.050	1.058
0.797	0.804	0.834	0.909	0.904	0.968	0.985	0.994	1.035	1.083
0.782	0.809	0.846	0.865	0.907	0.961	0.976	1.011	1.014	1.066
0.767	0.805	0.841	0.868	0.909	0.941	0.975	1.007	1.100	1.049
0.791	0.811	0.836	0.873	0.917	0.952	0.979	1.008	1.086	1.065
0.763	0.812	0.844	0.852	0.895	0.945	0.977	1.007	1.065	1.056
0.779	0.812	0.836	0.852	0.894	0.945	0.976	1.018	1.043	1.066
0.771	0.817	0.835	0.855	0.896	0.984	0.990	1.011	1.099	1.058
0.788	0.806	0.859	0.884	0.904	0.950	0.982	1.034	1.081	1.073
0.769	0.807	0.839	0.864	0.887	0.948	0.970	1.006	1.075	1.058
0.786	0.801	0.839	0.919	0.911	0.958	0.965	0.998	1.040	1.065
0.784	0.800	0.840	0.959	0.890	0.945	1.002	1.038	1.057	1.063
0.785	0.803	0.839	0.919	0.895	0.947	0.982	1.004	1.111	1.068
0.786	0.811	0.858	0.908	0.894	0.937	0.986	1.019	1.092	1.062
0.782	0.810	0.831	0.904	0.876	0.956	0.989	0.999	1.053	1.072
0.766	0.801	0.845	0.910	0.882	0.952	1.001	1.013	1.071	1.050
0.765	0.811	0.852	0.904	0.897	0.950	1.019	1.003	1.067	1.064
0.794	0.808	0.835	0.926	0.877	0.959	0.995	1.007	1.072	1.062
0.778	0.796	0.844	0.939	0.905	0.956	1.045	1.020	1.036	1.079
0.784	0.806	0.838	0.905	0.891	0.942	1.029	1.008	1.068	1.080
0.781	0.810	0.830	0.880	0.890	0.941	0.977	1.010	1.038	1.069
0.767	0.797	0.828	0.865	0.890	0.944	1.001	1.015	1.055	1.077
0.789	0.808	0.848	0.871	0.897	0.939	0.999	1.003	1.049	1.104
0.785	0.804	0.849	0.892	0.895	0.953	1.034	1.022	1.038	1.083
0.774	0.812	0.832	0.906	0.880	0.944	1.037	1.003	1.057	1.076
0.782	0.817	0.839	0.923	0.917	0.952	1.011	1.009	1.059	1.065
0.773	0.815	0.835	0.913	0.898	0.939	0.988	1.002	1.040	1.075
0.766	0.823	0.849	0.951	0.901	0.939	0.999	1.004	1.059	1.070
0.765	0.815	0.879	0.958	0.885	0.956	0.994	1.010	1.045	1.093
0.769	0.799	0.864	0.886	0.897	0.932	1.003	1.000	1.054	1.075
0.781	0.826	3.061	0.860	0.896	0.973	1.001	1.015	1.037	1.072

4 Przypadek Testowy 3-Algorytm 2-opt: Zależność wyników od danych początkowych

4.1 Cel:

Algorytm 2-opt bazuje na początkowej, ustalonej permutacji i na jej podstawie próbuje uzyskać lepszy wynik poprzez inwersję odpowiednich przeobrażeń. Celem tej części będzie sprawdzenie, jak duży wpływ ma dobór początkowej permutacji do otrzymanych wyników oraz jaki ma to wpływ na czas wykonywania algorytmu

4.2 Założenia:

Do tego badania użyto automatycznie wygenerowanych grafów typu **EUC2D**. Rozmiary grafów (oznaczane literą n) należą do zbioru $n \in \{10, 15, 20, \dots, 100\}$. Do testów przygotowano 4 wartości początkowe:

- Permutację losową
- Permutację otrzymaną przez uruchomienie algorytmu najbliższego sąsiada na wylosowanym wierzchołku
- Permutację otrzymaną przez uruchomienie rozszerzonego algorytmu najbliższego sąsiada
- Ciąg $i \in \{1, 2, \dots, n\}$

4.3 Wyniki:

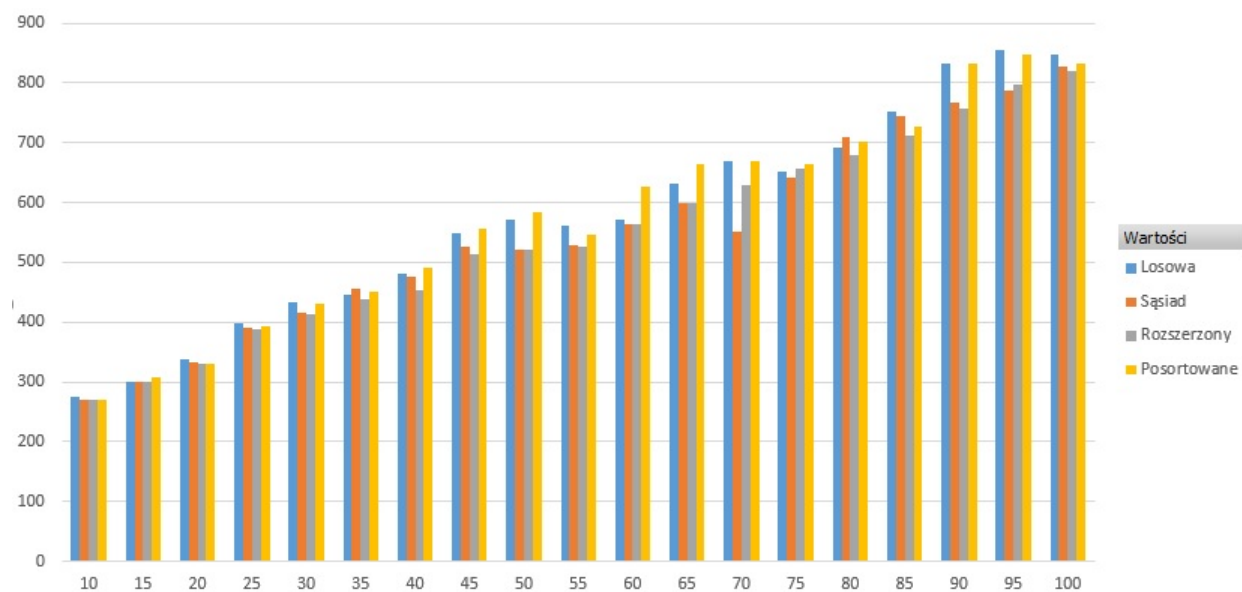
n	Losowa	Sąsiad	Rozszerzony	Podstawowa
10	276	271	271	271
15	300	300	300	308
20	337	332	330	330
25	399	391	387	392
30	434	416	414	430
35	445	457	439	450
40	482	476	453	490
45	550	527	513	556
50	572	521	522	584
55	562	528	526	547
60	571	563	563	626
65	631	598	598	663
70	670	552	630	668
75	651	642	656	663
80	691	709	680	703
85	751	744	712	726
90	833	766	757	833
95	855	786	798	848
100	848	827	821	832

Tabela 5: Funkcje celu dla zadanych grafów przy odpowiednich warunkach początkowych.

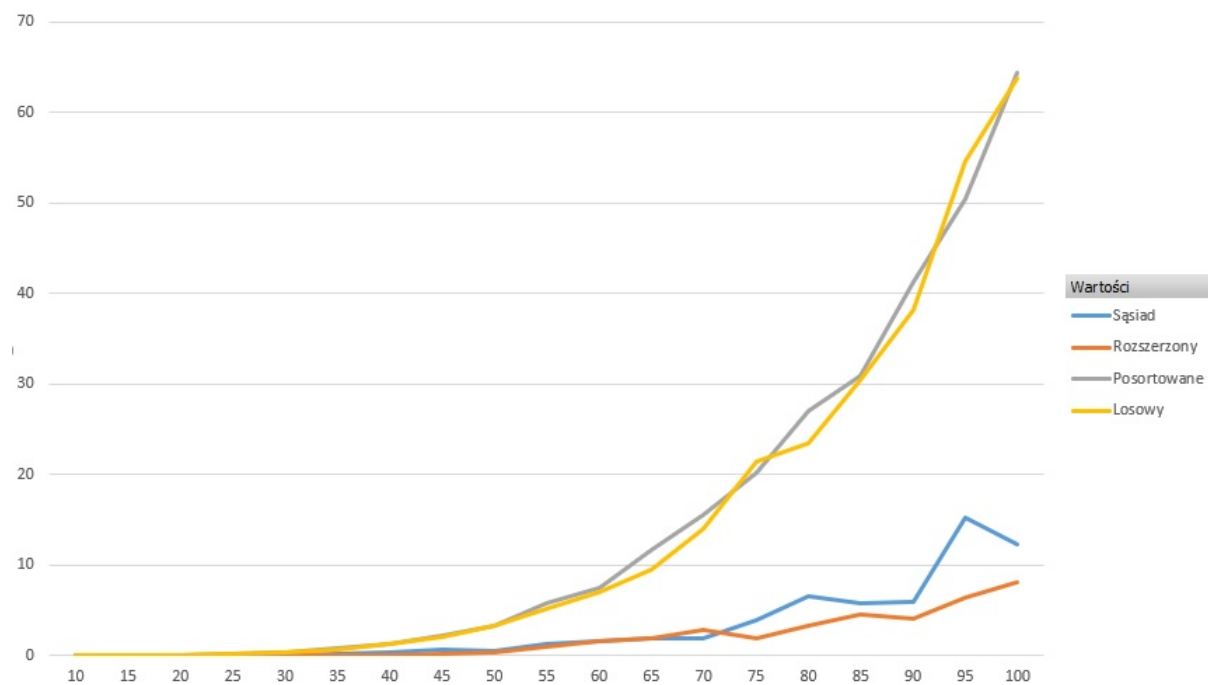
n	Losowa	Sąsiad	Rozszerzony	Podstawowa
10	0.001	0.001	0.0007	0.004
15	0.015	0.006	0.002	0.025
20	0.072	0.025	0.016	0.073
25	0.142	0.042	0.031	0.201
30	0.350	0.036	0.035	0.363
35	0.720	0.258	0.058	0.756
40	1.230	0.411	0.081	1.314
45	2.106	0.705	0.234	2.274
50	3.211	0.473	0.417	3,319
55	5,068	1,194	0,994	5,723
60	7,075	1,531	1,544	7,512
65	9,536	1,887	1,883	11,659
70	3,957	1,922	2,811	15,471
75	21,451	3,926	1,874	20,245
80	23,508	6,597	3,218	27,021
85	30,458	5,715	4,584	30,862
90	38,122	5,982	4,108	41,256
95	4,621	15,187	6,415	50,381
100	63,692	12,296	8,039	64,398

Tabela 6: Czas wykonania algorytmów dla zadanych grafów przy odpowiednich warunkach początkowych.

4.4 Wykresy:



Rysunek 4: Funkcja celu dla 2-opt dla różnych startowych permutacji



Rysunek 5: Czas dla 2-opt dla różnych startowych permutacji

4.5 Wnioski:

Mimo podobnych wyników dla wszystkich permutacji startowych zauważamy, iż w rankingu czasowym bardzo dobrze sprawdzają się permutacje, które wynikają z działania algorytmu najbliższego sąsiada. Dla początkowych permutacji w formie posortowanego ciągu lub wygenerowanych losowo uważamy złożoność ponadliniową.

5 przypadek testowy 4

5.1 Cel:

Celem badania jest sprawdzenie czy rzeczywista złożoność czasowa dla algorytmu two-opt jest zgodna z oczekiwanym $O(n^3)$.

5.2 Założenia:

Podczas badania korzystać będziemy z losowych instancji grafów pełnych, symetrycznych o $n \in \{30, 40, 50 \dots 100\}$ wierzchołkach. Badanie czasu każdego rozmiaru grafów zostało powtórzone 30 razy.

5.3 Wyniki:

Tabela czasów przedstawia czasy wykonania programu. Dane w tabach, dla czytelności, zostały zaokrąglone do 3 miejsc po przecinku. W dalszych obliczeniach korzystaliśmy z danych dokładnych zamieszczonych w pliku csv/xlsx.

30	40	50	60	70	80	90	100
0.479	0.645	0.889	1.399	2.497	4.054	6.243	8.768
0.477	0.629	0.889	1.397	2.494	3.746	6.227	8.748
0.478	0.662	0.884	1.400	2.473	3.726	6.265	8.781
0.473	0.641	0.881	1.388	2.506	3.768	6.229	8.796
0.472	0.645	0.889	1.398	2.457	3.697	6.254	8.921
0.480	0.641	0.885	1.397	2.469	3.713	6.253	8.829
0.474	0.637	0.885	1.400	2.457	3.688	6.238	8.791
0.477	0.640	0.886	1.406	2.467	3.724	6.211	8.824
0.479	0.647	0.882	1.545	2.629	3.687	6.264	8.759
0.471	0.637	0.884	1.508	2.504	3.666	6.294	8.732
0.490	0.639	0.889	1.466	2.608	3.738	6.386	8.696
0.479	0.639	0.882	1.473	2.458	3.763	6.405	8.770
0.481	0.642	0.892	1.477	2.464	3.816	6.356	8.747
0.476	0.639	0.884	1.460	2.461	3.720	6.398	8.875
0.480	0.642	0.883	1.459	2.465	3.743	6.442	8.904
0.476	0.640	0.903	1.471	2.514	3.714	6.347	8.988
0.477	0.644	0.878	1.472	2.458	3.762	6.451	8.958
0.464	0.639	0.889	1.468	2.485	3.759	6.272	8.829
0.496	0.637	0.886	1.539	2.445	3.754	6.491	8.984
0.477	0.642	0.880	1.478	2.452	3.720	6.261	8.798
0.474	0.639	0.882	1.469	2.458	3.729	6.367	8.872
0.476	0.644	0.884	1.472	2.458	3.784	6.423	8.805
0.611	0.641	0.884	1.462	2.438	3.804	6.218	8.701
0.472	0.638	0.883	1.447	2.449	3.679	6.197	8.705
0.480	0.645	0.895	1.454	2.856	3.617	6.245	8.872
0.472	0.641	0.897	1.461	2.449	3.697	6.195	8.894
0.473	0.641	0.873	1.485	2.453	3.659	6.261	8.802
0.485	0.639	0.898	1.474	2.435	3.657	6.280	8.900
0.478	0.642	0.880	1.515	2.462	3.670	6.302	8.923
0.471	0.642	0.888	1.600	2.455	3.761	6.310	8.987

Tabela 7: tabela czasów dla poszczególnych n, czasy podane w sekundach

n	30	40	50	60	70	80	90
T	0.482	0.641	0.886	1.461	2.489	3.734	6.303
SD	0.025	0.005	0.006	0.050	0.082	0.076	0.083
SE	0.005	0.001	0.001	0.009	0.015	0.014	0.015

Tabela 8: T - średni czas wykonania (w sekundach), SD - odchylenie standardowe, SE - błąd standardowy

Odchylenie standardowe oraz błąd standardowy zostały obliczone według wzorów:

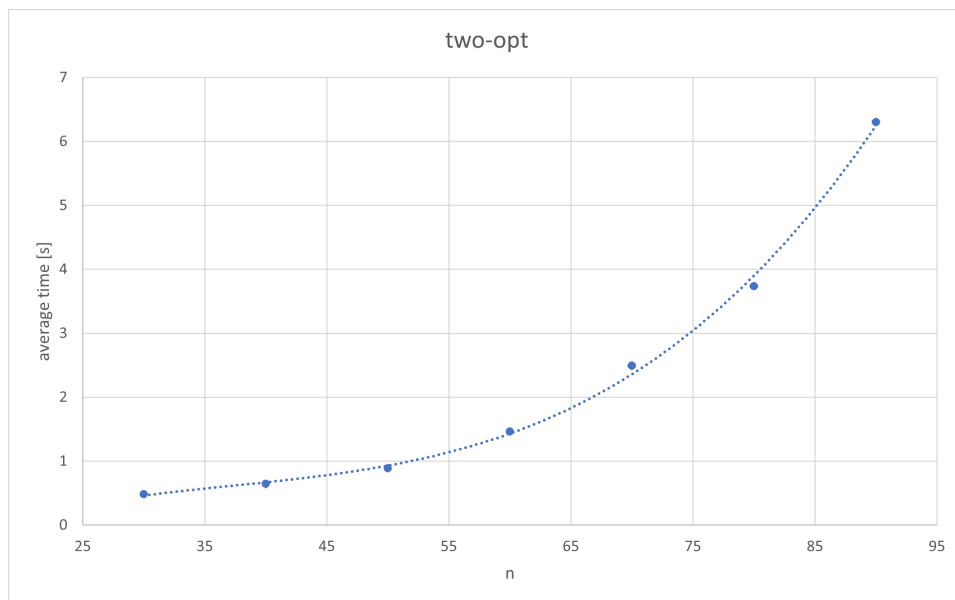
Odchylenie standardowe:

$$\sigma = \sqrt{\frac{\sum_{n=1}^{100} (\bar{x} - x_n)^2}{100}}$$

Błąd standardowe:

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{100}}$$

5.4 Wykresy:



Rysunek 6: średnie czasy wykonania wraz z błędami standardowymi

Na osi OX wykresu naniesione zostały n dla których wykonywany był pomiar, na osi OY uśrednione wartości czasów wykonania wraz z zaznaczonymi błędami standardowymi.

5.5 Wnioski:

Zmierzone dane potwierdziły tezę, że złożoność czasowa metody two-opt ustalonym jest $O(n^3)$.

6 przypadek testowy 5

6.1 Cel:

Celem badania jest sprawdzenie jak dokładne są rozwiązania metodą two-opt dla instancji testowych problemów symetrycznych i asymetrycznych z [TSPLIB](#).

6.2 Założenia:

Podczas badania korzystać będziemy z wybranych instancji symetrycznych oraz asymetrycznych z biblioteki TSPLIB w trzech formatach danych FULL MATRIX, LOWER DIAG ROW i EUC 2D. Najlepszym wynikiem - referencyjnym będziemy określać najmniejszą długość ścieżki w grafie wyznaczoną przez jedną z trzech algorytmów: k-random, nearest-neighbor, two-opt.

6.3 Wyniki:

instance name	two-opt cost	reference cost	PRD
a280.tsp	2715	2715	0
bays29.tsp	2108	2108	0
berlin52.tsp	8384	8384	0
bier127.tsp	123545	123545	0
ch130.tsp	6457	6457	0
ch150.tsp	7056	7053	0.042535091
d198.tsp	16340	16340	0
d493.tsp	37645	37645	0
dantzig42.tsp	699	699	0
eil51.tsp	443	443	0
eil76.tsp	570	570	0
eil101.tsp	662	662	0
fl417.tsp	12188	12188	0
fri26.tsp	937	937	0
gil262.tsp	2601	2601	0
gr17.tsp	2211	2211	0
gr21.tsp	2801	2801	0
gr24.tsp	1278	1278	0
gr48.tsp	5278	5278	0
gr120.tsp	7390	7390	0
hk48.tsp	11718	11718	0
kroA100.tsp	22883	22883	0
kroA150.tsp	28600	28600	0
kroA200.tsp	31148	31148	0
kroB100.tsp	23134	23134	0
kroB150.tsp	28223	28223	0
kroB200.tsp	31767	31767	0
kroC100.tsp	22727	22727	0
kroD100.tsp	23218	23218	0
lin105.tsp	14941	14941	0
lin318.tsp	45721	45721	0
pr76.tsp	121207	121207	0
pr107.tsp	47691	46927	1.628060605
pr124.tsp	63212	63212	0
pr136.tsp	99770	99770	0
pr144.tsp	58796	58796	0
pr152.tsp	75662	75662	0

Tabela 9: TSP

instance name	two-opt cost	reference cost	PRD
br17.atsp	39	39	0
ft53.atsp	8657	8657	0
ft70.atsp	44669	43264	3.247503698
ftv170.atsp	5140	3959	29.83076534
ftv33.atsp	2036	1664	22.35576923
ftv35.atsp	2245	1869	20.11771001
ftv38.atsp	2264	1830	23.71584699
ftv44.atsp	2522	1884	33.8641189
ftv47.atsp	3136	2338	34.13173653
ftv55.atsp	2907	2344	24.01877133
ftv64.atsp	4173	2629	58.72955496
ftv70.atsp	3434	2715	26.4825046
kro124p.atsp	52705	44612	18.14085896
p43.atsp	5638	1023	451.1241447
rbg323.atsp	4251	1742	144.0298507
rbg358.atsp	4649	1806	157.4197121
rbg403.atsp	5207	3543	46.96584815
rbg443.atsp	5897	3913	50.70278559
ry48p.atsp	15965	15965	0

Tabela 10: aTSP

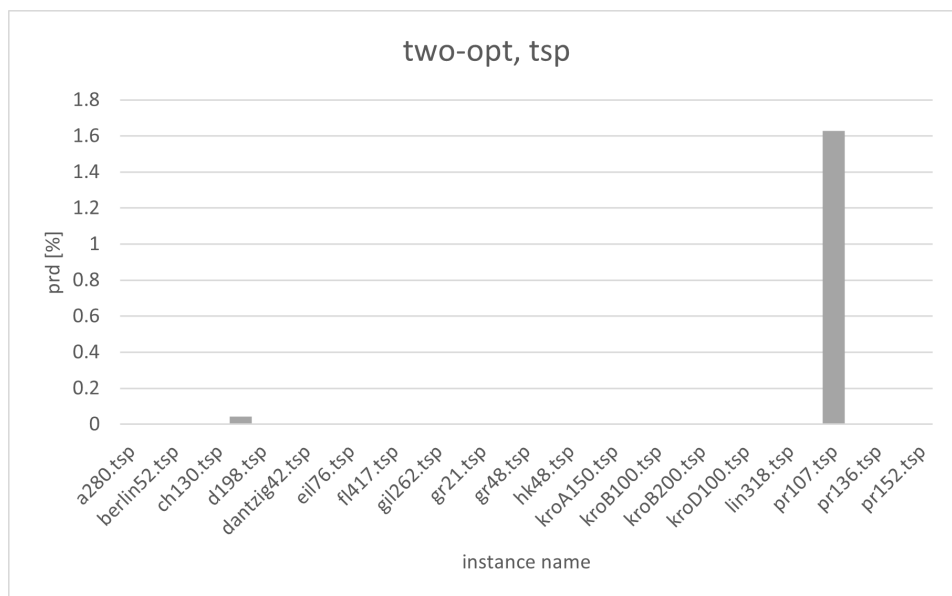
W tabelach przedstawione zostały wyniki eksperymentu oraz obliczone na ich podstawie PRD według wzoru

$$PRD_x = 100 \cdot \frac{x - x_{ref}}{x_{ref}}$$

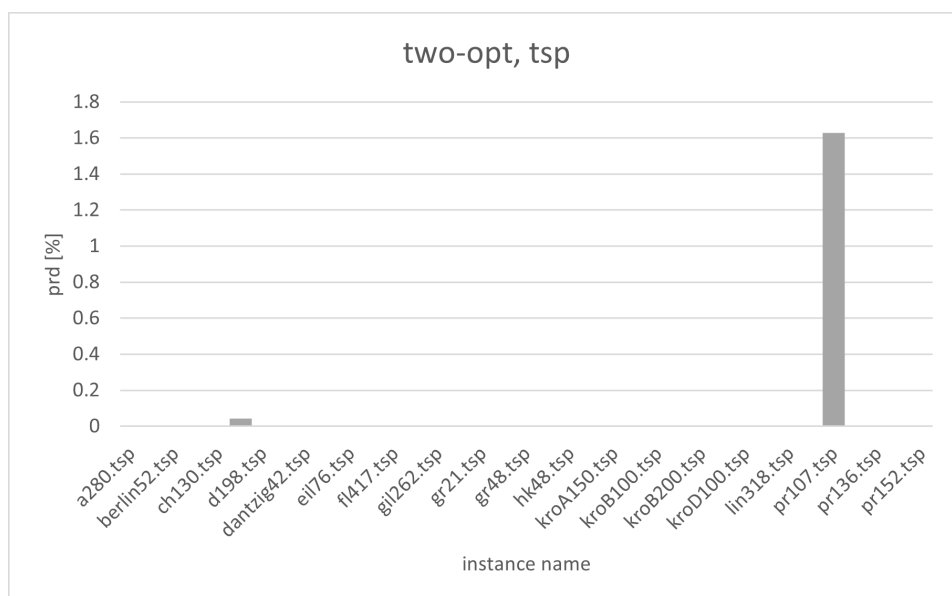
wariant	tsp	atsp
średnie PRD	0.045151235	60.25670957
odchylenie standardowe	0.267548599	103.9669193

Tabela 11: średnie PRD dla obu wariantów

6.4 Wykresy:



Rysunek 7: Wartości PRD dla instancji TSP



Rysunek 8: Wartości PRD dla instancji ATSP

Na osi OX wykresu naniesione zostały nazwy instancji dla których wykonywany był pomiar, na osi OY zostały naniesione wartości PRD.

6.5 Wnioski:

Uzyskane dane sugerują, że algorytm two-opt dużo lepiej radzi sobie z instancjami symetrycznymi problemów. Otóż może być to część prawdy, gdyż w instancjach asymetrycznych nie są (nie muszą być) zachowane nierówności trójkąta. Z drugiej strony 2 instancje symetryczne w których algorytm nie dał najlepszego znanego wyniku miały postać euklidesową gdzie nierówności trójkąta były zachowane.

7 Testy statystyczne: EUC2D dla 2-opt i rozszerzonego sąsiada

7.1 Cel:

Bazując na wiedzy teoretycznej wiemy, iż 2-opt dla rozkładu miast spełniającego nierówność trójkątą powinien otrzymywać lepsze wyniki niż rozszerzony algorytm najbliższego sąsiada, ze względu na to, iż potrafi on dokonać przestawienia wierzchołków, aby nie dochodziło do potencjalnych przecięć na trasie. Celem tego etapu jest sprawdzenie, czy ta hipoteza jest zgodna z rzeczywistością.

7.2 Założenia:

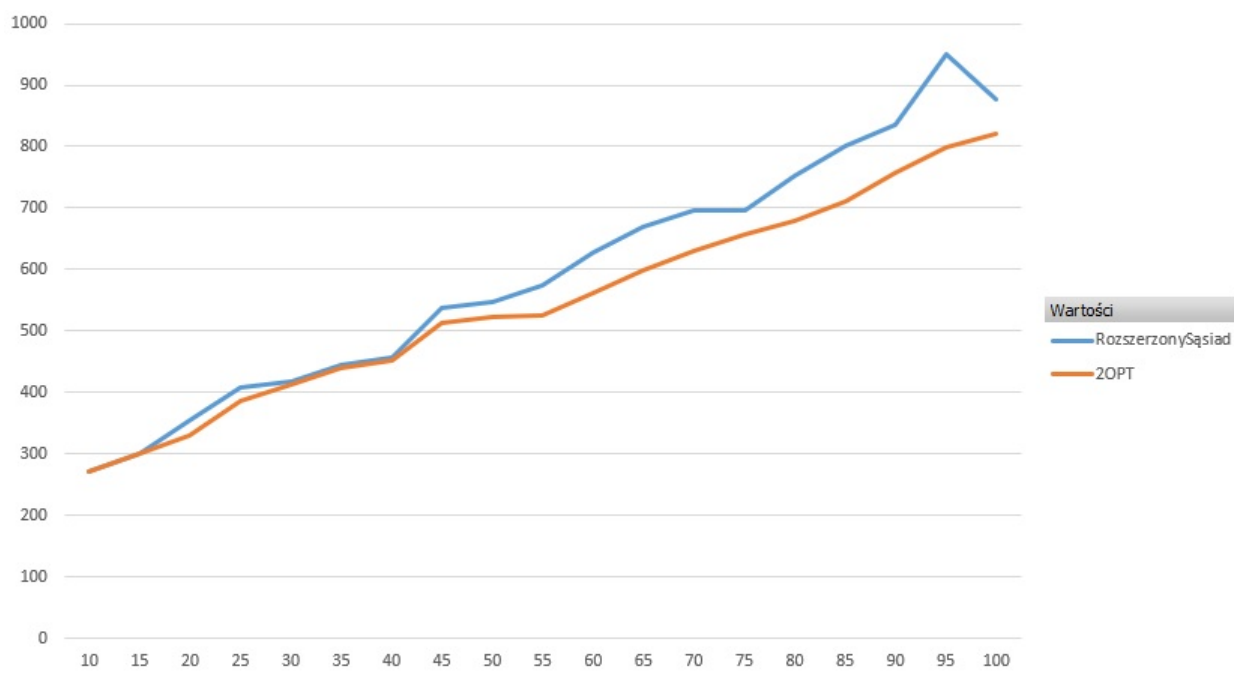
Do tego badania użyto automatycznie wygenerowanych grafów typu **EUC2D**. Rozmiary grafów (oznaczane literą n) należą do zbioru $n \in \{10, 15, 20, \dots, 100\}$. Dla algorytmu 2-opt element startowy został automatycznie wygenerowany.

7.3 Wyniki:

n	RozszerzonySąsiad	2-OPT
10	271	271
15	300	300
20	354	330
25	408	387
30	419	414
35	444	439
40	458	453
45	537	513
50	548	522
55	573	526
60	627	563
65	670	598
70	697	630
75	696	656
80	752	680
85	801	712
90	835	757
95	949	798
100	877	821

Tabela 12: Funkcje celu dla wybranych algorytmów.

7.4 Wykresy:



Rysunek 9: Funkcja celu dla 2-opt dla różnych startowych permutacji

7.5 Test statystyczny Wilcoxona:

Hipoteza zerowa: $OPT \leq ENN$, gdzie OPT oznacza algorytm 2-opt, a ENN- rozszerzony algorytm najbliższego sąsiada.

Hipoteza alternatywna: $OPT > ENN$

$$W_- = 153.$$

$$W_+ = 0.$$

7.6 Wnioski:

Wartość krytyczna $\alpha = 0.05$. Dla tego typu statystyk $T_{crit} = 41$ (dana z tabeli dla hipotez "o jednym ogonie"). Hipoteza zerowa jest odrzucona, gdy $T \leq 41$. U nas $T = 0$, jako minimum z W_- i W_+ , więc hipoteza zerowa zostaje odrzucona. Jest wystarczająco powodów na to, aby uważać, iż różnica median jest większa od 0.

Pair	OPT	ENN	Abs.Diff	Rank	Sign
5	414	419	5	2	-1
6	439	444	5	2	-1
7	453	458	5	2	-1
4	387	408	21	4	-1
3	330	354	24	5.5	-1
8	513	537	24	5.5	-1
9	522	548	26	7	-1
14	656	696	40	8	-1
10	526	573	47	9	-1
11	563	627	64	11	-1
13	630	697	67	12	-1
12	598	670	72	13.5	-1
15	680	752	72	13.5	-1
17	757	835	78	15	-1
16	712	801	89	16	-1
18	798	949	151	17	-1

Tabela 13: Tabela rang dla testu Wilcoxona

8 Testy statystyczne: FULLMATRIX dla 2-opt i rozszerzonego sąsiada

8.1 Cel:

Typ danych FULLMATRIX zmienia zasady gry, gdyż w tego typu przykładach nierówność trójkątą nie musi być zachowana. Celem tego etapu jest sprawdzenie reakcji algorytmów na nowy typ danych

8.2 Założenia:

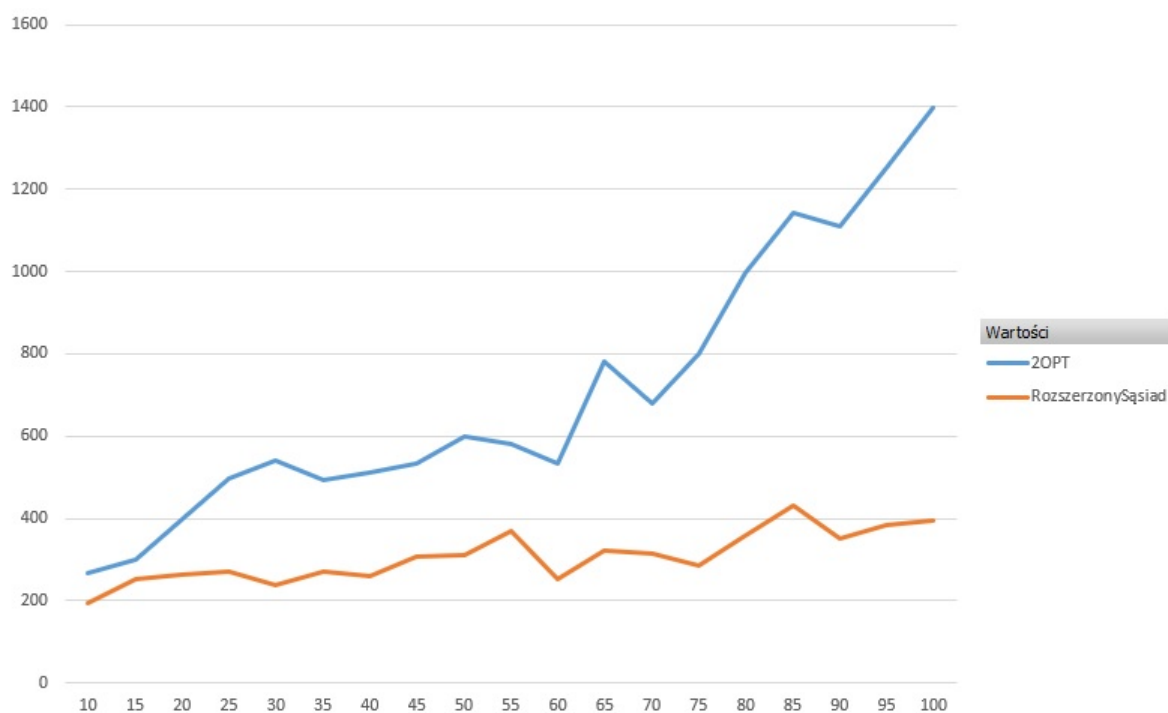
Do tego badania użyto automatycznie wygenerowanych grafów typu **FULL MATRIX**. Rozmiary grafów (oznaczane literą n) należą do zbioru $n \in \{10, 15, 20, \dots, 100\}$. Dla algorytmu 2-opt element startowy został automatycznie wygenerowany.

8.3 Wyniki:

n	2-OPT	RozszerzonySąsiad
10	266	193
15	300	252
20	400	265
25	498	271
30	541	239
35	492	272
40	512	261
45	534	306
50	600	310
55	580	370
60	534	252
65	780	323
70	678	316
75	800	286
80	995	357
85	1141	432
90	1110	352
95	1251	384
100	1399	396

Tabela 14: Funkcje celu dla wybranych algorytmów.

8.4 Wykresy:



Rysunek 10: Funkcja celu dla 2-opt dla różnych startowych permutacji

8.5 Test statystyczny Wilcoxona:

Hipoteza zerowa: $OPT \leq ENN$, gdzie OPT oznacza algorytm 2-opt, a ENN- rozszerzony algorytm najbliższego sąsiada.

Hipoteza alternatywna: $OPT > ENN$

$$W_- = 0.$$

$$W_+ = 190.$$

8.6 Wnioski:

Wartość krytyczna $\alpha = 0.05$. Dla tego typu statystyk $T_{crit} = 53$ (dana z tabeli dla hipotez "o jednym ogonie"). Hipoteza zerowa jest odrzucona, gdy $T \leq 53$. U nas $T = 0$, jako minimum z W_- i W_+ , więc hipoteza zerowa zostaje odrzucona.

Pair	OPT	ENN	Abs.Diff	Rank	Sign
2	300	252	48	1	+1
1	266	193	73	2	+1
3	400	265	135	3	+1
10	580	370	210	4	+1
6	492	272	220	5	+1
4	498	271	227	6	+1
8	534	306	228	7	+1
7	512	261	251	8	+1
11	534	252	282	9	+1
9	600	310	290	10	+1
5	541	239	302	11	+1
13	678	316	362	12	+1
12	780	323	457	13	+1
14	800	286	514	14	+1
15	995	357	638	15	+1
16	1141	432	709	16	+1
17	1110	352	758	17	+1
18	1251	384	867	18	+1
19	1399	396	1003	19	+1

Tabela 15: Tabela rang dla testu Wilcoxona