

Algorytmy Metaheurystyczne 1

Adrian Herda

8 stycznia 2024

1 Wyniki

nazwa pliku	najlepsze rozwiązanie	MST	Local search na bazie DFS MST			Losowy local search			Przyspieszony losowy local search		
			avg kroki	avg waga	min waga	avg kroki	avg waga	min waga	avg kroki	avg waga	min waga
xqf131	564	474	15.9	649.77	658	133.72	612.46	582	124.61	981.18	813
xqg237	1019	897	36.38	1170.26	1166	261.54	1117.13	1066	245.06	1944.00	1523
pma343	1368	1179	43.14	1561.10	1588	404.46	1483.75	1417	395.05	2586.37	2138
pka379	1332	1151	61.94	1587.48	1574	450.21	1447.03	1383	438.60	2534.46	2041
bcl380	1621	1444	42.57	1958.13	1976	449.25	1818.05	1726	412.20	3442.95	2754
pbl395	1281	1124	43.72	1584.38	1637	459.17	1427.61	1359	426.74	2652.45	2156
pbk411	1343	1180	38.57	1638.37	1623	485.06	1492.49	1435	447.42	2829.54	2202
pbn423	1365	1201	37.43	1688.24	1714	498.31	1522.15	1445	465.91	2828.98	2325
pbm436	1443	1269	46.83	1730.79	1776	513.56	1610.40	1529	473.36	3039.58	2425
xql662	2513	2240	62.72	3164.32	3191	812.53	2813.24	2708	771.82	5270.68	4347
xit1083	3558	3253	87.05	4396.00	4440	1386.61	4019.99	3919	1330.18	7811.51	6595
icw1483	4416	4015	96.75	5693.67	5723	1930.10	4982.06	4868	1864.24	9745.39	8407
djc1785	6115	5541	99.97	8035.09	8027	2352.39	6881.12	6742	2285.75	13097.22	11461
djb2036	6197	5593	100.0	9682.48	9633	2723.75	7009.80	6852	2658.48	13797.59	11841
pds2566	7643	6956	100.0	10178.85	10136	3481.51	8713.09	8550	3041.39	17456.76	15112

Jak możemy zauważyć, zarówno lokalne przeszukiwanie rozpoczynające się od losowej permutacji, jak i lokalne przeszukiwanie rozpoczynające się od cyklu utworzonego z MST (Minimalnego Drzewa Rozpinającego), skutkowały dobrą przybliżoną wartością, z niewielką przewagą dla drugiego podejścia. Przyspieszenie lokalnego przeszukiwania skutkowało gorszymi wynikami; było szybkie, ale nadal wymagało wielu kroków, podczas gdy lokalne przeszukiwanie oparte na MST było szybkie, wymagało rzędu mniejszej liczby kroków i dawało najlepsze rezultaty. Ważne jest wybranie dobrego punktu startowego dla metaheurystyki.

2 Pytania z listy

2.1 Dlaczego w przypadku metryki euklidesowej ten algorytm tworzy cykl bez przecinających się krawędzi?

Ponieważ inwersja dotyczy rozplątywania przecięć. I w ten sposób zawsze skracamy długość cyklu.

2.2 Dlaczego więc nie jest tak w metryce z której korzystamy?

Mimo że rozplątane linie mają mniejszą długość w metryce euklidesowej, zaokrąglone przecinające się krawędzie mają również długości równe jedności.

Przykładem może być kwadrat, o boku równym 1, którego przekątne w metryce euklidesowej będą w przybliżeniu równe 1,41. W metryce, którą używamy przekątne te przybliżą się do wartości $1,41 \approx 1$, a więc do wartości takiej samej jak bok. z tego powodu nie warto jest zmieniać tych krawędzi i tracić moc obliczeniową.

