

Algorytmy Metaheurystyczne Komiwojażer Heurystycznie

Gabriel Budziński(254609)
Franciszek Stepek (256310)

Przedmowa

Sprawko na metaheurystykę

1 Podsumowanie złożoności obliczeniowych implementacji

1.1 K-Random

Co nieco o nim

1.2 Nearest Neighbor

Co nieco o nim

1.3 Nearest Branching Neighbor

Co nieco o nim

1.4 2-Opt

Co nieco o nim

1.5 3-Opt

Co nieco o nim

2 Opis eksperymentów

2.1 Implementacja

C/Cpp

2.2 Sprzęt

16GB Ramu

2.2.1 Pececik

6 rdzonków

2.2.2 Lapek

4 rdzonki

2.3 Instancje

2.3.1 Przykładziki ze stronki

2.3.2 Instancje losowe

2.4 Metodologia/cel

2.5 Opis wyników

Testy Wilcoxona dla poszczególnych algorytmów:

- W pierwszej dużej tabeli będziemy mieli zestawienie 3 algorytmów
- W drugiej i trzeciej (Już nieco 'ciekawszej') Zobaczymy jak na rozwiązanie *2-Opt* wpływają warunki początkowe (zarówno czasowo, jak i w kontekście funkcji Celu)
- W czwartej i piątej otrzymamy większe zestawienie różnych przypadków dla *Nearest Neighbor*

2.5.1 Część I

Tabela numer 1:

Nr. Instancji	1	2	3	4	5	1-2	1-3	4-5
1	3157	32621	2846	2922	33913	-29464	311	-30991
2	8980	29990	8385	8746	25809	-21010	595	-17063
3	135737	610252	126892	128097	622778	-474515	8845	-494681
4	7579	47194	7098	6741	49975	-39615	481	-43234
5	8191	53545	7157	7372	54720	-45354	1034	-47348
6	803	3509	714	677	3560	-2706	89	-2883
7	511	1586	454	443	1750	-1075	57	-1307
8	642	2606	561	595	2645	-1964	81	-2050
9	27807	168501	23737	23006	172442	-140694	4070	-149436
10	33633	270344	28612	29756	242459	-236711	5021	-212703
11	35859	331888	31749	34109	345599	-296029	4110	-311490
12	29158	154024	25393	25014	174054	-124866	3765	-149040
13	34499	256477	28435	29312	253834	-221978	6064	-224522
14	36980	338421	33375	34698	339346	-301441	3605	-304648
15	26227	174278	23107	22850	178774	-148051	3120	-155924
16	26947	173366	24721	25011	174970	-146419	2226	-149959
17	27460	170965	24459	23810	154305	-143505	3001	-130495
18	20356	126612	18281	15972	121878	-106256	2075	-105906
19	54019	593697	49141	46246	606262	-539678	4878	-560016
20	54019	582949	49141	45758	575474	-528930	4878	-529716
21	68964	1405305	60930	62216	1415877	-1336341	8034	-1353661
22	331103	6434381	282758	279968	6392975	-6103278	48345	-6113007
23	46680	597495	45340	49462	572145	-550815	1340	-522683
24	69297	735352	61910	64640	677100	-666055	7387	-612460
25	120769	842650	110103	107601	882692	-721881	10666	-775091
26	61652	821860	61400	68116	823348	-760208	252	-755232
27	85699	988886	79996	81923	1028352	-903187	5703	-946429
28	94683	1638917	87542	88659	1635424	-1544234	7141	-1546765
29	58023	1137733	56765	55439	1183487	-1079710	1258	-1128048
30	59890	766136	52403	56468	791951	-706246	7487	-735483
31	131281	1940053	118468	119774	1962350	-1808772	12813	-1842576
32	153462	586378	121315	121297	568936	-432916	32147	-447639
33	2752	22583	2497	2684	22420	-19831	255	-19736
34	8605	110285	7566	7628	114431	-101680	1039	-106803
35	11054	183204	9415	9627	182635	-172150	1639	-173008
36	1554	8755	1418	1346	8903	-7201	136	-7557
37	830	3585	770	775	3466	-2755	60	-2691
38	152493	1629073	139697	140164	1576951	-1476580	12796	-1436787
39	5030	41053	4418	4294	42391	-36023	612	-38097

Przy czym kolejne kolumny 'numeryczne' oznaczają:

- 1 - Wartość funkcji Celu dla algorytmu *Nearest Neighbor* startującego z wybranego 'miasta'
- 2 - Wartość funkcji Celu dla algorytmu *K-Random* działającego tak długo, jak z kolumny 1
- 3 - Wartość funkcji Celu dla algorytmu *2-Opt* startującego z rozwiązania w punkcie 1
- 4 - Wartość funkcji Celu dla algorytmu *2-Opt* startującego z losowej instancji
- 5 - Wartość funkcji Celu dla algorytmu *K-Random* działającego tak długo, jak z kolumny 4

A teraz jak rozumieć kolejne kolumny 'różnicowe':

- 1-2 - Porównanie działania KR z NN dla tego samego budżetu obliczeniowego
- 1-3 - 'Dowód', że 2O rzeczywiście poprawia rozwiązanie startowe

- 4-5 - Porównanie działania KR z 2O dla tego samego budżetu obliczeniowego

Zauważmy, że we wszystkich tych kolumnach każda z wartości jest tego samego znaku, zatem bez dokładniejszej analizy można powiedzieć, że wartość statystyki testowej dla testu Wilcozona będzie równa 0, zatem jednoznacznie powie nam, który z algorytmów zwraca zawsze lepsze rozwiązanie. Pojawia się nam zatem następująca zależność (gdzie znak ' $<$ ' oznacza, że lewa wartość zwraca nam 'gorsze' rozwiązanie od prawego):

$$K\text{-Random} < \text{Nearest Neighbor} < 2\text{-Opt}$$

2.5.2 Część II

Tabela numer 2:

Nr. Instancji	NN	KR	NN-KR	NN-KR	Rangi
1	2846	2992	-146	146	11
2	8385	7967	418	418	16
3	126892	124006	2886	2886	29
4	7098	6988	110	110	8
5	7157	7080	77	77	6.5
6	714	689	25	25	4
7	454	461	-7	7	2
8	561	567	-6	6	1
9	23737	24667	-930	930	19
10	28612	29780	-1168	1168	21
11	31749	32634	-885	885	18
12	25393	24066	1327	1327	22
13	28435	29226	-791	791	17
14	33375	33294	81	81	7
15	23107	22870	237	237	13
16	24721	23678	1043	1043	20
17	24459	24275	184	184	12
18	18281	15962	2319	2319	26
19	49141	46422	2719	2719	28
20	49141	45633	3508	3508	32
21	60930	61190	-260	260	14
22	282758	287980	-5222	5222	34
23	45340	52126	-6786	6786	38
24	61910	67957	-6047	6047	36
25	110103	103341	6762	6762	37
26	61400	66836	-5436	5436	35
27	79996	82941	-2945	2945	30
28	87542	90788	-3246	3246	31
29	56765	58313	-1548	1548	24
30	52403	56508	-4105	4105	33
31	118468	119808	-1340	1340	23
32	121315	118891	2424	2424	27
33	2497	2624	-127	127	10
34	7566	7643	-77	77	6.5
35	9415	9687	-272	272	15
36	1418	1399	19	19	3
37	770	799	-29	29	5
38	139697	137507	2190	2190	25
39	4418	4531	-113	113	9

Przy czym kolejne kolumny oznaczają:

- NN - Wartość funkcji Celu dla algorytmu *2-Opt* startującego z rozwiązania znalezione przez algorytm *Nearest Neighbor*
- KR - Wartość funkcji Celu dla algorytmu *2-Opt* startującego z rozwiązania znalezione przez algorytm *K-Random* (Startowe rozwiązanie w tym samym czasie, co startowe dla NN)
- Kolejne 2 kolumny znaczą dokładnie to, co ich nazwa, natomiast ostatnia wyznacza rangi dla testu Wilcoxona

A teraz policzmy jeszcze tylko 2 sumy dla Wilcoxona:

- $T_+ = 315.5$
- $T_- = 432.5$

Zatem nasza statystyka końcowa będzie wynosiła 315.5. Trzeba tu teraz coś konstruktywnie dopowiedzieć, dlaczego to oznacza, że nie ma w sumie za bardzo znaczenia z czego startuje...

Tabela numer 3:

Nr. Instancji	NN	KR	NN-KR
1	0.115728	3.10199	-2.986262
2	0.00120772	0.00922684	-0.00801912
3	0.0187193	0.145382	-0.1266627
4	0.0138299	0.184798	-0.1709681
5	0.0393003	0.338091	-0.2987907
6	0.00987954	0.0552343	-0.04535476
7	0.00119699	0.0103625	-0.00916551
8	0.00364293	0.0333502	-0.02970727
9	0.0154935	0.137051	-0.1215575
10	0.053393	0.313875	-0.260482
11	0.144667	0.836886	-0.692219
12	0.0138314	0.0914137	-0.0775823
13	0.037316	0.41158	-0.374264
14	0.109228	0.997454	-0.888226
15	0.0119046	0.102974	-0.0910694
16	0.01084	0.0888192	-0.0779792
17	0.0122842	0.114743	-0.1024588
18	0.0165922	0.0705111	-0.0539189
19	0.322236	4.21766	-3.895424
20	0.343797	2.56742	-2.223623
21	27.5015	656.948	-629.4465
22	11.1163	179.428	-168.3117
23	0.0065809	0.110337	-0.1037561
24	0.0144687	0.196237	-0.1817683
25	0.0155958	0.195165	-0.1795692
26	0.00640445	0.246059	-0.23965455
27	0.0186396	0.358017	-0.3393774
28	0.062336	0.966489	-0.904153
29	0.0857857	2.09062	-2.0048343
30	0.221278	4.1955	-3.974222
31	0.673865	13.1858	-12.511935
32	0.00617383	0.0256009	-0.01942707
33	0.046445	0.83168	-0.785235
34	1.95213	32.5976	-30.64547
35	4.56492	81.1747	-76.60978
36	0.0119262	0.103982	-0.0920558
37	0.00323845	0.0298058	-0.02656735
38	0.0786318	0.723296	-0.6446642
39	0.0935837	1.57451	-1.4809263

Oczywiście ta tabela jest analogiczna do poprzedniej, ale tym razem zamiast wartości Celu mamy czas działania samej części *2-Opt*.

Jak widać, wszystkie różnice są ujemne, ale (w połączeniu z danymi otrzymanymi w tabeli 1) można powiedzieć, że łatwo uzależnić czas działania algorytmu 2-Opt od pierwotnego rozwiązania - jeżeli jest znacznie 'lepsze', to także widać, że różnice w czasie są znacznie wyższe.

Uwaga na boku: wydaje mi się, że lepiej będzie to przenieść do poprzedniej tabeli i zrobić p prostu zbiorcze podsumowanie

2.5.3 Część III

Tabela numer 4:

Nr. Instancji	1N	FN	1B	FB
1	3157	2975	3452	2998
2	8980	8181	8980	8181
3	135737	133953	129421	128279
4	7579	7129	7198	6908
5	8191	7113	8191	7113
6	803	746	804	748
7	511	482	545	499
8	642	608	621	599
9	27807	24698	26854	24815
10	33633	31479	33612	31479
11	35859	34543	35794	34543
12	29158	25884	29158	25884
13	34499	31611	32825	31611
14	36980	35389	36980	35389
15	26227	23660	26227	23564
16	26947	24852	26947	24852
17	27460	24782	27460	24782
18	20356	16935	20356	16935
19	54019	49201	54019	49201
20	54019	49201	54019	49201
21	68964	68531	70315	67864
22	331103	313745	322807	310967
23	46680	46680	49166	47899
24	69297	67055	71550	67323
25	120769	114553	120769	114553
26	61652	60964	61652	60964
27	85699	79564	85699	79564
28	94683	92552	94683	92903
29	58023	54491	58635	54604
30	59890	58279	58151	58151
31	131281	127230	131445	126062
32	153462	130921	153462	130921
33	2752	2612	2632	2606
34	8605	7993	8700	7934
35	11054	10540	10763	10657
36	1554	1437	1544	1432
37	830	796	801	751
38	152493	140486	152493	140486
39	5030	4578	4941	4691

Przy czym kolejne kolumny oznaczają:

- 1N - Wartość funkcji Celu dla algorytmu *Nearest Neighbor* startującego z wybranego miasta
- FN - Wartość funkcji Celu dla algorytmu *Nearest Neighbor* startującego z każdego miasta, i wybierający najlepsze
- 1B - Wartość funkcji Celu dla algorytmu *Branching Nearest Neighbor* (z głębokością = 2) startującego z wybranego miasta
- FB - Wartość funkcji Celu dla algorytmu *Branching Nearest Neighbor* (z głębokością = 2) startującego z każdego miasta, i wybierający najlepsze

Tabela numer 4:

Oczywiście ta tabela jest analogiczna do poprzedniej, ale tym razem zamiast wartości Celu mamy czas działania.

Uwaga ogólna: tutaj warto by się w sumie chwile zastanowić, co z czym porównywać, ale początek już jakiś jest...

Nr. Instancji	1N	FN	1B	FB
1	0.000644468	0.0983119	0.0048885	1.01056
2	2.07E-05	0.00103294	3.76E-05	0.00205635
3	0.000146889	0.0121529	0.000282864	0.0361393
4	0.000104299	0.0139546	0.000241166	0.0310817
5	0.000134752	0.0224471	0.000239433	0.0382092
6	6.75E-05	0.00723383	0.000454636	0.0460773
7	2.06E-05	0.00110371	0.000153318	0.00747459
8	6.48E-05	0.0034514	0.000231878	0.0173178
9	6.99E-05	0.00736386	0.00012696	0.0131532
10	0.000222951	0.0240762	0.00021405	0.0365781
11	0.000252144	0.0555201	0.000354758	0.0790306
12	9.39E-05	0.00853953	0.000109656	0.0112842
13	0.000131333	0.0222619	0.000253897	0.0373115
14	0.000236748	0.0511064	0.000379147	0.0778277
15	6.77E-05	0.00713671	0.000131215	0.0138976
16	0.000125232	0.00788403	0.000222792	0.0115013
17	6.63E-05	0.00694331	0.000201731	0.0129614
18	5.77E-05	0.00701994	0.000113084	0.0166362
19	0.000421003	0.166132	0.000678634	0.260951
20	0.000420879	0.155087	0.000638746	0.240323
21	0.0102591	14.3142	0.0157081	24.5947
22	0.0042375	4.41593	0.0108562	11.6176
23	5.57E-05	0.00624229	0.000471479	0.0663509
24	8.92E-05	0.00915955	0.000177797	0.0226451
25	8.82E-05	0.012921	0.000207303	0.0303293
26	9.07E-05	0.0139036	0.000168004	0.0304266
27	0.000102496	0.0173458	0.000168221	0.0335902
28	0.000204923	0.0498144	0.00163079	0.352259
29	0.000284032	0.0803066	0.000632295	0.193229
30	0.000401408	0.125099	0.00113758	0.380743
31	0.000791101	0.383126	0.0016203	0.873932
32	3.31E-05	0.00248934	6.17E-05	0.00508362
33	0.000155953	0.0351781	0.000429553	0.113054
34	0.00137766	0.857353	0.0040416	2.48614
35	0.00274642	2.70393	0.00916338	7.70218
36	5.00E-05	0.00510014	0.000238881	0.022591
37	7.09E-05	0.00241989	0.000155393	0.0108239
38	0.000206588	0.049608	0.00139074	0.239691
39	0.000318213	0.0560495	0.000668361	0.163452

2.6 Wnioski

Drobne uwagi