List 3 report

Albert Kołodziejski

December 22, 2023

Results:

data	best	Local search		Simulated annealing		Tabu search	
name	solution	avg cost	min cost	avg cost	min cost	avg cost	min cost
XQF131	564	612.46	582	617.65	589	613.47	586
XQG237	1019	1117.13	1066	1139.50	1083	1116.57	1058
PMA343	1368	1483.75	1417	1523.87	1452	1485.36	1424
PKA379	1332	1447.03	1383	1443.44	1393	1446.61	1391
BCL380	1621	1818.05	1726	1832.54	1722	1818.53	1745
PBL395	1281	1427.61	1359	1434.57	1371	1429.24	1371
PBK411	1343	1492.49	1435	1508.28	1444	1490.75	1431
PBN423	1365	1522.15	1445	1531.47	1443	1523.87	1452
PBM436	1443	1610.40	1529	1617.39	1540	1611.68	1548
XQL662	2513	2813.24	2708	2831.25	2702	2817.85	2682
XIT1083	3558	4019.99	3919	4051.19	3948	4018.64	3905
ICW1483	4416	4982.06	4868	4983.27	4857	4987.55	4868
DJC1785	6115	6881.12	6742	6930.86	6801	6877.29	6741
DJB2036	6197	7009.80	6852	7012.32	6837	7030.27	6873
PDS2566	7643	8713.09	8550	8670.99	8525	8699.68	8504

In every run, both Simulated annealing and Tabu search were run with the same parameters. That means for some Data sets, Local search gives the best, for some Simulated annealing and for some Tabu search. That's why we have to fine-tune our metaheuristic for our data.

Implementations:

Simulated annealing:

Generally, I wanted to make everything dependent on the size of the data which I will call from now on n, so after some trial and error I came up with those numbers:

- initial temperature: $n/2$
- colling rate: 0.95
- epoch length: n
- max iterations without improvement: n

Tabu search:

Generally, I wanted to make everything dependent on the size of the data which I will call from now on n, so after some trial and error I came up with those numbers:

- use of tabu list with 7 last moves
- use of aspiration criterion
- looks every time for the best neighbor
- max iterations without improvement: n