Report 4

Candidate moves

Evolutionary Computation

Zuzanna Buchnajzer 148280

Algorithm

The extension of the local search algorithm was initiated with a random solution. The neighborhood was composed of steepest versions of firstly external node exchange, and then internal edge swap. The steepest version fully searches the space of candidate moves for each of the nodes. To ensure that for each node there is at least one feasible candidate move, the algorithm terminates the search when at least one candidate move is feasible for a given swap, and after consideration of at least 10 candidate vertices.

Remarks

During optimizing the code for the candidate moves, I discovered some bugs, regarding traversing the nested for loops in the edge swap function. After resolving it, the best algorithm (random_steepest_edges) gets slightly better results regarding quality and time.

Results

The results are obtained by aggregating the total cost of algorithms started from random solutions. The time is the sum of these 200 algorithm runs for a given configuration in seconds. The table presents the previous (not fixed experiments), random_steepest_edges without applying candidate moves and after applying.
As expected the results after limiting the space of the possible moves, slightly worse, however, the time saved by this modification is in my opinion worth this sacrifice.

main_key	min	max	mean	time
random_greedy_nodes	53256,00	72111,00	61635,20	207,82
random_greedy_edges	71202,00	88144,00	77969,92	133,70
random_steepest_nodes	57938,00	74022,00	63986,65	1866,64
random_steepest_edges	47715,00	65163,00	52822,70	952,11
best_greedy_nodes	50008,00	59918,00	54477,65	420,45
best_greedy_edges	49588,00	59456,00	54431,23	435,60
best_steepest_nodes	49980,00	59544,00	54213,37	589,86
best_steepest_edges	50117,00	59952,00	54524,07	445,73
r_s_e_no_candidate	45782,00	52750,00	48324,61	633,37
r_s_e_candidate	46684,00	53727,00	50063,02	47,33

Repository link: https://github.com/Zinath09/EC-course/tree/lab4