INF273 – Assignment 3

Lukas Schramm 22nd February 2022

1 Functions

Currently, I am maintaining the following helper functions in my Utils file.[1]

- load_problem: Given a path to a file, it reads the content of the file into a dictionary of information.
- feasibility_check: It takes a solution (list) and a problem dictionary and checks if the solution is feasible. If it is not feasible, it outputs the reason why. It does not check validity. This one got changed after Ahmed found a major mistake in it. I hope it now works correctly
- cost_function: It takes a solution (list) and a problem dictionary and calculates the cost of the function. As feasibility_check it does not check if the original solution was valid.
- splitting_a_list_at_zeros: Helper function which splits a solution into vehicles and if needed a dummy vehicle.
- initial_solution: Generates an initial default solution to start with. This is always a solution where the dummy vehicle handles all calls.
- random_solution: Generates a random solution. The generator itself is quite bad in my view because I overtuned it a bit. It automatically gives one vehicle exactly one call and the rest goes to the dummy vehicle. That way I got solutions for file 3 and 4 but the solutions for all files are quite bad.[2]
- blind_random_search: Takes a problem and a number of iterations to find the best out of n random feasible solutions if any is found.
- blind_search_latex_generator: This function runs the blind_random_search and writes the data into LaTeXtables since I am obviously too lazy to do it myself.
- latex_add_line: Adds a new result line into an results table of this file.
- latex_replace_line: Change the optimal solution and its seed in that file.

  If there are any questions or nice recommendations to get a better structure, just send me a message.

[1] The green ones are changes or additions from the last assignment

[2] But since we do not need that random solution generator any longer I keep it like that.

Moreover, there is a new file for Heuristics where I collect all of the important algorithms and their helper functions.

- alter_solution_1insert: A function which takes a current solution and outputs a next solution by using the 1-insert-operation. The output is not necessary feasible, but of course valid
- alter_solution_2exchange: A function which takes a current solution and outputs a next solution by using the 2-exchange-operation. The output is not necessary feasible, but of course valid
- alter_solution_3exchange: A function which takes a current solution and outputs a next solution by using the 3-exchange-operation. The output is not necessary feasible, but of course valid
- local_search: This function takes a problem, an initial solution, a number of iterations (10.000) and the allowed neighbouring function and performs a local search
- simulated_annealing: This function takes a problem, an initial solution, a number of iterations (10.000) and the allowed neighbouring function and performs a simulated annealing
- local_search_sim_annealing_latex: This function takes as input the allowed neighbouring function(s), the heuristics method, the problem and the number of iterations and performs the heuristics on randomly chosen seeds. It then calculates the average time and objective and runs the LaTeXfunctions to change the tables of this PDF

2 Result tables

Table 1: Call_7_Vehicle_3

Method	Average objective	Best objective	Improvement (%)	Running time
Random search	2289893.35	2120884	34.59%	0.62s
Local Search-1-insert	1287284.00	1134176	65.02%	0.86s
Local Search-2-exchange	1215881.20	1134176	65.02%	0.85s
Local Search-3-exchange	1324206.50	1134176	65.02%	0.81s
Simulated Annealing-1-insert	1254356.20	1134176	65.02%	1.09s
Simulated Annealing-2-exchange	1269867.50	1134176	65.02%	0.89s
Simulated Annealing-3-exchange	1344723.10	1134176	65.02%	0.79s

Listing 1: Optimal solution call_7_vehicle_3

```
sol = [4, 4, 7, 7, 0, 2, 2, 0, 1, 5, 5, 3, 3, 1, 0, 6, 6]
seeds = [123967717, 611293610, 947178443, 416487721, 625094973,
    248273999, 644655455, 809681614, 995086831, 842960241]
```

Table 2: Call_18_Vehicle_5

Method	Average objective	Best objective	Improvement (%)	Running time
Random search	7195792.08	6215552	29.80%	0.80s
Local Search-1-insert	3174322.30	2745474	69.36%	1.43s
Local Search-2-exchange	3031088.50	2452476	72.63%	1.35s
Local Search-3-exchange	3157474.90	2841718	68.28%	1.17s
Simulated Annealing-1-insert	3507291.20	2871979	67.95%	1.29s
Simulated Annealing-2-exchange	3086672.10	2466354	72.47%	1.45s
Simulated Annealing-3-exchange	3370478.80	2935212	67.24%	1.70s

Listing 2: Optimal solution call_18_vehicle_5

```
sol = [4, 4, 1, 15, 15, 16, 16, 1, 0, 6, 6, 5, 5, 11, 12, 11, 12, 0, 2,
    9, 2, 9, 18, 18, 13, 13, 0, 7, 7, 14, 14, 0, 3, 3, 10, 10, 17, 17, 0,
     8, 8]
seeds = [227124072, 905011295, 709745415, 61191232, 837331316,
    821298357, 431635577, 417757480, 692892166, 639256496]
```

Table 3: Call_35_Vehicle_7

Method	Average objective	Best objective	Improvement (%)	Running time
Random search	15924073.22	14436028	20.19%	1.09s
Local Search-1-insert	7790461.70	7305006	60.27%	1.54s
Local Search-2-exchange	7650079.50	6596229	64.13%	1.39s
Local Search-3-exchange	7866091.20	7079307	61.50%	1.35s
Simulated Annealing-1-insert	7377672.50	6929461	62.31%	1.21s
Simulated Annealing-2-exchange	7711098.90	6918732	62.37%	1.23s
Simulated Annealing-3-exchange	8223222.70	7530283	59.05%	1.27s

Listing 3: Optimal solution call_35_vehicle_7

```
sol = [15, 15, 16, 16, 7, 7, 33, 33, 0, 9, 14, 9, 14, 35, 35, 21, 21, 0,
     23, 23, 12, 17, 17, 27, 27, 22, 12, 22, 0, 34, 34, 6, 6, 13, 13, 25,
     25, 0, 19, 19, 10, 28, 10, 32, 28, 29, 32, 29, 20, 20, 0, 11, 30,
    11, 30, 18, 5, 18, 2, 5, 2, 31, 31, 0, 1, 1, 0, 3, 3, 4, 4, 8, 8, 24,
     24, 26, 26]
seeds = [416638676, 613072689, 899001257, 995439543, 475898410,
    802204519, 526284085, 591102712, 83139499, 218479724]
```

Table 4: Call_80_Vehicle_20

Method	Average objective	Best objective	Improvement (%)	Running time
Random search	39584864.24	37697832	18.28%	2.44s
Local Search-1-insert	18087011.80	17358268	62.89%	3.59s
Local Search-2-exchange	18258089.40	16587155	64.53%	2.67s
Local Search-3-exchange	18415634.40	16925173	63.81%	2.53s
Simulated Annealing-1-insert	17830017.80	16597079	64.51%	3.06s
Simulated Annealing-2-exchange	18025991.00	17262516	63.09%	2.64s
Simulated Annealing-3-exchange	19129494.30	17271853	63.07%	2.46s

Listing 4: Optimal solution call_80_vehicle_20

```
sol = [4, 4, 71, 71, 3, 3, 0, 22, 61, 22, 55, 55, 28, 28, 61, 0, 54, 63,
     54, 63, 42, 42, 0, 15, 15, 29, 29, 67, 17, 17, 73, 73, 10, 10, 67,
    0, 59, 59, 7, 50, 7, 14, 14, 50, 0, 53, 11, 11, 44, 53, 44, 5, 5, 0,
    1, 68, 34, 1, 68, 34, 33, 33, 0, 35, 62, 35, 62, 77, 77, 0, 21, 21,
    58, 58, 0, 66, 66, 80, 80, 16, 16, 0, 51, 27, 27, 51, 0, 23, 9, 23,
    79, 69, 69, 9, 72, 79, 72, 0, 41, 41, 57, 57, 37, 31, 31, 37, 6, 6,
    0, 65, 36, 65, 24, 36, 56, 56, 24, 0, 74, 46, 74, 75, 46, 75, 0, 38,
    38, 26, 48, 26, 48, 0, 70, 32, 70, 45, 45, 32, 12, 12, 0, 40, 64, 64,
     40, 47, 47, 0, 39, 39,
      49, 49, 2, 2, 52, 52, 0, 60, 18, 60, 18, 78, 78, 0, 8, 8, 13, 13,
         19, 19, 20, 20, 25, 25, 30, 30, 43, 43, 76, 76]
seeds = [167203283, 810923845, 755688758, 769857558, 176670237,
    912492327, 583230321, 104086031, 747282985, 663554436]
```

Table 5: Call_130_Vehicle_40

Method	Average objective	Best objective	Improvement (%)	Running time
Random search	76627567.00	76627567	0.00%	4.52s
Local Search-1-insert	27087163.60	25235911	67.07%	7.40s
Local Search-2-exchange	28712355.10	27140349	64.58%	5.83s
Local Search-3-exchange	29880009.40	28846093	62.36%	4.82s
Simulated Annealing-1-insert	28095197.40	26270306	65.72%	6.11s
Simulated Annealing-2-exchange	28456448.80	26797867	65.03%	5.40s
Simulated Annealing-3-exchange	30121731.00	26886341	64.91%	5.42s

Listing 5: Optimal solution call_130_vehicle_40

```
sol = [66, 66, 0, 60, 60, 25, 25, 46, 46, 0, 33, 33, 18, 37, 37, 18, 0,
    123, 1, 123, 1, 0, 99, 99, 10, 10, 0, 107, 107, 109, 91, 91, 109, 0,
    96, 96, 68, 68, 0, 5, 5, 22, 22, 79, 79, 20, 20, 30, 30, 0, 88, 88,
    70, 28, 28, 70, 0, 106, 89, 89, 106, 0, 32, 69, 69, 32, 127, 127, 0,
    15, 15, 53, 53, 0, 121, 121, 17, 17, 12, 12, 0, 42, 42, 26, 26, 0,
    23, 59, 23, 59, 0, 114, 113, 113, 35, 114, 35, 0, 3, 3, 95, 29, 29,
    95, 0, 103, 103, 118, 118, 111, 56, 111, 56, 0, 73, 55, 55, 92, 73,
    92, 0, 102, 102, 112, 112, 51, 51, 64, 64, 0, 115, 129, 129, 115, 0,
    98, 8, 98, 8, 0, 72, 130, 130, 72, 76, 76, 0, 38, 38, 4,
       4, 31, 31, 0, 75, 82, 82, 75, 81, 81, 0, 21, 21, 45, 63, 65, 45,
        63, 65, 0, 124, 124, 9, 9, 0, 80, 80, 24, 24, 0, 27, 44, 44,
        27, 122, 47, 47, 122, 71, 71, 0, 117, 117, 39, 77, 77, 39, 0,
        119, 119, 116, 116, 36, 36, 0, 16, 16, 54, 54, 125, 101, 101,
        125, 0, 11, 41, 41, 11, 43, 43, 0, 126, 58, 58, 126, 7, 7, 14,
        14, 0, 50, 62, 62, 78, 78, 50, 0, 108, 108, 52, 52, 57, 57, 0,
        34, 120, 34, 120, 83, 48, 48, 83, 0, 74, 74, 104, 40, 104, 40,
        0, 97, 90, 97, 94, 90, 94, 128, 128, 0, 100, 67, 100, 61, 61,
        67, 110, 110, 0, 2, 2, 6, 6, 13, 13, 19, 19, 49, 49, 84, 84,
        85, 85, 86, 86, 87, 87, 93, 93, 105, 105]
seeds = [110144783, 678719075, 74993939, 266545900, 268617326,
    423635654, 679597528, 708635133, 758051865, 284753079]
```

Table 6: Call_300_Vehicle_90

Method	Average objective	Best objective	Improvement (%)	Running time
Random search	170784643.00	170784643	0.00%	10.66s
Local Search-1-insert	67963584.80	65285939	61.77%	14.77s
Local Search-2-exchange	73199763.10	68225050	60.05%	13.25s
Local Search-3-exchange	81482110.50	79014826	53.73%	14.81s
Simulated Annealing-1-insert	68267033.60	65105394	61.88%	13.29s
Simulated Annealing-2-exchange	73234328.20	70017120	59.00%	13.00s
Simulated Annealing-3-exchange	80613543.60	76781634	55.04%	14.45s

Listing 6: Optimal solution call_300_vehicle_90

```
sol = [94, 94, 165, 165, 0, 278, 278, 112, 112, 0, 263, 263, 286, 74,
    286, 74, 0, 98, 98, 294, 95, 95, 294, 0, 14, 186, 14, 186, 0, 96, 96,
     240, 240, 0, 68, 155, 68, 155, 0, 300, 77, 77, 125, 300, 56, 125,
    56, 0, 271, 212, 271, 212, 0, 126, 126, 253, 253, 187, 187, 0, 270,
    183, 270, 183, 0, 118, 118, 287, 287, 11, 11, 0, 235, 235, 231, 231,
    0, 250, 148, 250, 148, 0, 182, 55, 182, 55, 0, 288, 199, 288, 199, 0,
     269, 269, 227, 154, 154, 227, 0, 254, 51, 51, 254, 0, 281, 281, 139,
     139, 0, 277, 277, 0, 225, 284, 284, 225, 283, 283, 0, 213, 213, 289,
     289, 0, 26, 26, 298, 298, 0, 110, 110, 21, 21, 0, 222, 222, 164,
    164, 0, 25, 25, 275, 275, 0, 35, 273, 273, 35, 44, 40,
       44, 40, 99, 99, 0, 280, 280, 297, 146, 297, 140, 140, 146, 0, 111,
             111, 17, 17, 0, 65, 209, 65, 209, 80, 80, 0, 188, 188, 181,
            220, 181, 220, 202, 202, 0, 261, 257, 257, 261, 0, 233, 233,
            127, 127, 115, 115, 0, 201, 41, 210, 41, 201, 210, 0, 194, 163,
             194, 12, 163, 12, 0, 248, 248, 124, 124, 106, 106, 0, 36, 88,
            36, 91, 88, 91, 22, 22, 0, 158, 158, 72, 131, 72, 131, 198,
            198, 0, 159, 159, 60, 60, 0, 37, 224, 37, 224, 52, 52, 104,
            104, 0, 109, 109, 145, 145, 0, 206, 61, 206, 61, 0, 264, 219,
            264, 219, 0, 246, 203, 246, 189, 203, 189, 0, 71, 71, 132, 132,
             0, 79, 79, 20, 20, 0, 32, 89, 89, 32, 101, 101, 0, 83, 83,
            130, 130, 0, 234, 234, 16, 16, 0, 293, 293, 63, 184, 184, 63,
            0, 29, 29, 81, 157, 157, 81, 0, 245, 245, 113, 113, 0, 82, 49,
            49, 82, 0, 5, 5, 34, 50, 50, 34, 0, 114, 114, 226, 226, 102,
            102, 0, 62, 291, 122, 62, 291, 122, 0, 279, 279, 262, 262, 0,
            272, 272, 258, 258, 8, 8, 0, 67, 295, 185, 67, 295, 185, 156,
            156, 0, 218, 218, 243, 243, 46, 46, 292, 292, 0, 58, 90, 58,
            90, 0, 116, 116, 216, 216, 70, 2, 2, 70, 0, 193, 193, 252, 252,
             0, 59, 84, 59, 84, 0, 172, 249, 172, 249, 117, 117, 0, 241,
            214, 211, 211, 241, 214, 0, 53, 3, 3, 192, 192, 53, 0, 200,
            200, 9, 27, 9, 27, 0, 171, 171, 221, 221, 97, 97, 0, 30, 204,
            204, 196, 30, 129, 129, 196, 0, 64, 236, 236, 64, 0, 69, 191,
            191, 69, 0, 228, 228, 73, 73, 0, 103, 103, 4, 4, 0, 207, 45,
            268, 255, 45, 268, 207, 153, 255, 153, 0, 167, 151, 167, 197,
            151, 197, 100, 100, 0, 142, 176, 176, 142, 0, 168, 168, 256,
            173, 256, 173, 0, 247, 247, 149, 42, 149, 38, 42, 38, 0, 24,
            24, 296, 296, 0, 237, 237, 85, 85, 0, 134, 134, 107, 107, 0,
            128, 128, 282, 282, 0, 123, 123, 7, 7, 170, 170, 0, 92, 92,
            119, 119, 0, 23, 23, 141, 120, 120, 141, 0, 19, 144, 19, 144,
            0, 179, 1, 121, 179, 121, 1, 0, 15, 195, 15, 290, 290, 195, 0,
            266, 251, 266, 251, 0, 6, 6, 10, 10, 13, 13, 18, 18, 28, 28,
            31, 31, 33, 33, 39, 39, 43, 43, 47, 47, 48, 48, 54, 54, 57, 57,
             66, 66, 75, 75, 76, 76, 78, 78, 86, 86, 87, 87, 93, 93, 105,
            105, 108, 108, 133, 133, 135, 135, 136, 136, 137, 137, 138,
            138, 143, 143, 147, 147, 150, 150, 152, 152, 160, 160, 161,
            161, 162, 162, 166, 166, 169, 169, 174, 174, 175, 175, 177,
            177, 178, 178, 180, 180, 190, 190, 205, 205, 208, 208, 215,
            215, 217, 217, 223, 223, 229, 229, 230, 230, 232, 232, 238,
            238, 239, 239, 242, 242, 244, 244, 259, 259, 260, 260, 265,
            265, 267, 267, 274, 274, 276, 276, 285, 285, 299, 299]
seeds = [562466833, 287198709, 807067543, 773873945, 959281844,
    723614494, 635225881, 1637614, 965454146, 737549028]
```