INF273 – Assignment 1

Lukas Schramm 11th February 2022

1 Comments on my code

I am programming with `Python` and rewrote the framework we were given but basically followed it's concept. I have written several automatic tests checking if my feasibility-, cost- and reading functions are working and outputing the same results. They all do, but that does not necessarily mean that my code is 100% code (which makes me nervous). I aim to continue to write automatic tests since I have planned to learn myself testing for ages.

1.1 Functions

Currently, I am maintaining the following functions.

- 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 (since the aim
- 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.
- 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.[1]
- 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.

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

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

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

Listing 1: Optimal solution call_7_vehicle_3

```
sol = [5, 5, 0, 7, 7, 0, 1, 1, 0, 4, 4, 6, 6, 2, 2, 3, 3]
seeds = [863843277, 415483601, 100086270, 533050748, 347542105, ↩
    418599890, 177232060, 565112754, 187975592, 466961181]
```

Table 2: Call_18_Vehicle_5

Method	Average objective	Best objective	Improvement (%)	Running time
Random search	7195792.08	6215552	29.80%	0.80s

Listing 2: Optimal solution call_18_vehicle_5

```
1 sol = [17, 17, 0, 8, 8, 0, 6, 6, 0, 1, 1, 0, 12, 12, 0, 10, 10, 11, ↩
     11, 18, 18, 13, 13, 7, 7, 2, 2, 9, 9, 14, 14, 16, 16, 5, 5, 15, 15,↩
      4, 4, 3, 3]
2 seeds = [591815520, 540747627, 127735185, 529643335, 38918856, ↩
     610354960, 37013615, 779714863, 126344857, 133121881]
```

Table 3: Call_35_Vehicle_7

Method	Average objective	Best objective	Improvement (%)	Running time
Random search	15924073.22	14436028	20.19%	1.09s

Listing 3: Optimal solution call_35_vehicle_7

```
1 sol = [1, 1, 0, 26, 26, 0, 32, 32, 0, 21, 21, 0, 31, 31, 0, 18, 18, 0,↩
      12, 12, 0, 35, 35, 23, 23, 8, 8, 30, 30, 9, 9, 27, 27, 15, 15, 29,↩
      29, 33, 33, 13, 13, 24, 24, 22, 22, 7, 7, 16, 16, 25, 25, 5, 5, ↩
     14, 14, 17, 17, 20, 20, 2, 2, 19, 19, 3, 3, 34, 34, 10, 10, 28, 28,↩
      4, 4, 11, 11, 6, 6]
2 seeds = [621663642, 715692516, 268263705, 383418018, 951124417, ↩
     243073611, 610063942, 649573103, 341301930, 38686832]
```

Table 4: Call_80_Vehicle_20

Method	Average objective	Best objective	Improvement (%)	Running time
Random search	39584864.24	37697832	18.28%	2.44s

Listing 4: Optimal solution call_80_vehicle_20

```
sol = [37, 37, 0, 42, 42, 0, 63, 63, 0, 1, 1, 0, 14, 14, 0, 70, 70, 0,
    12, 12, 0, 49, 49, 0, 21, 21, 0, 53, 53, 0, 59, 59, 0, 7, 7, 0,
   73, 73, 0, 10, 10, 0, 43, 43, 0, 32, 32, 0, 36, 36, 0, 80, 80, 0,
   35, 35, 0, 61, 61, 0, 13, 13, 74, 74, 31, 31, 40, 40, 78, 78, 68,
   68, 19, 19, 38, 38, 22, 22, 11, 11, 41, 41, 46, 46, 23, 23, 5, 5,
   55, 55, 52, 52, 75, 75, 79, 79, 2, 2, 17, 17, 67, 67, 20, 20, 69,
   69, 56, 56, 60, 60, 64, 64, 66, 66, 77, 77, 76, 76, 50, 50, 45, 45,
    47, 47, 6, 6, 16, 16, 34, 34, 72, 72, 58, 58, 3, 3, 65, 65, 27,
   27, 4, 4, 26, 26, 48, 48, 25, 25, 29, 29,
sol = 18, 18, 62, 62, 39, 39, 15, 15, 44, 44, 30, 30, 9, 9, 57, 57,
   24, 24, 51, 51, 33, 33, 8, 8, 28, 28, 71, 71, 54, 54]
seeds = [20969160, 99643375, 441725204, 847317039, 441438469,
   252535348, 136552150, 506985547, 206312491, 874371276]
```

Table 5: Call_130_Vehicle_40

Method	Average objective	Best objective	Improvement (%)	Running time
Random search	76627567.00	76627567	0.00%	4.52s

Listing 5: Optimal solution call_130_vehicle_40

```
sol = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ↩
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 2, ↩
    2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10, 10, 11, 11, 12, ↩
    12, 13, 13, 14, 14, 15, 15, 16, 16, 17, 17, 18, 18, 19, 19, 20, 20,↩
     21, 21, 22, 22, 23, 23, 24, 24, 25, 25, 26, 26, 27, 27, 28, 28, ↩
    29, 29, 30, 30, 31, 31, 32, 32, 33, 33, 34, 34, 35, 35, 36, 36, 37,↩
     37, 38, 38, 39, 39, 40, 40, 41, 41, 42, 42, 43, 43, 44, 44, 45, ↩
    45, 46, 46, 47, 47, 48, 48, 49, 49, 50, 50, 51, 51, 52, 52, 53, 53,↩
     54, 54, 55, 55,
sol = 56, 56, 57, 57, 58, 58, 59, 59, 60, 60, 61, 61, 62, 62, 63, 63, ↩
    64, 64, 65, 65, 66, 66, 67, 67, 68, 68, 69, 69, 70, 70, 71, 71, 72,↩
     72, 73, 73, 74, 74, 75, 75, 76, 76, 77, 77, 78, 78, 79, 79, 80, ↩
    80, 81, 81, 82, 82, 83, 83, 84, 84, 85, 85, 86, 86, 87, 87, 88, 88,↩
     89, 89, 90, 90, 91, 91, 92, 92, 93, 93, 94, 94, 95, 95, 96, 96, ↩
    97, 97, 98, 98, 99, 99, 100, 100, 101, 101, 102, 102, 103, 103, ↩
    104, 104, 105, 105, 106, 106, 107, 107, 108, 108, 109, 109, 110, ↩
    110, 111, 111, 112, 112, 113, 113, 114, 114, 115, 115, 116, 116, ↩
    117, 117, 118, 118, 119, 119, 120, 120, 121, 121, 122, 122, 123, ↩
    123, 124, 124, 125, 125, 126, 126, 127, 127, 128, 128, 129, 129, ↩
    130, 130]
seeds = [85511460, 741472138, 280783314, 299190331, 673225625, ↩
    173219145, 691274920, 286652312, 780496710, 692814446]
```

Table 6: Call_300_Vehicle_90

Method	Average objective	Best objective	Improvement (%)	Running time
Random search	170784643.00	170784643	0.00%	10.66s

Listing 6: Optimal solution call_300_vehicle_90

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

```
    254, 254, 255, 255, 256, 256, 257, 257, 258, 258, 259, 259, 260, ↩
    260, 261, 261, 262, 262, 263, 263, 264, 264, 265, 265, 266, 266, ↩
    267, 267, 268, 268, 269, 269, 270, 270, 271, 271, 272, 272, 273, ↩
    273, 274, 274, 275, 275, 276, 276, 277, 277, 278, 278, 279, 279, ↩
    280, 280, 281, 281, 282, 282, 283, 283, 284, 284, 285, 285, 286, ↩
    286, 287, 287, 288, 288, 289, 289, 290, 290, 291, 291, 292, 292, ↩
    293, 293, 294, 294, 295, 295, 296, 296, 297, 297, 298, 298, 299, ↩
    299, 300, 300]
seeds = [675717729, 126893339, 433169188, 181659905, 747892400, ↩
    168687230, 569481925, 807648437, 64957997, 492307850]
```