Repetita results might be false:

When using the topology : “2016TopologyZooUCL_inverseCapacity/Airtel.graph”, the results from any of my algorithms and the 2-SR-MIP-NO-SPLIT from Repetita were different.

I only managed to obtain a maximum utilistaion of 1.375849, while the 2-SR-MIP-NO-SPLIT managed to obatin a maximum utilisation of 0.9000175 (on demand file 0000).

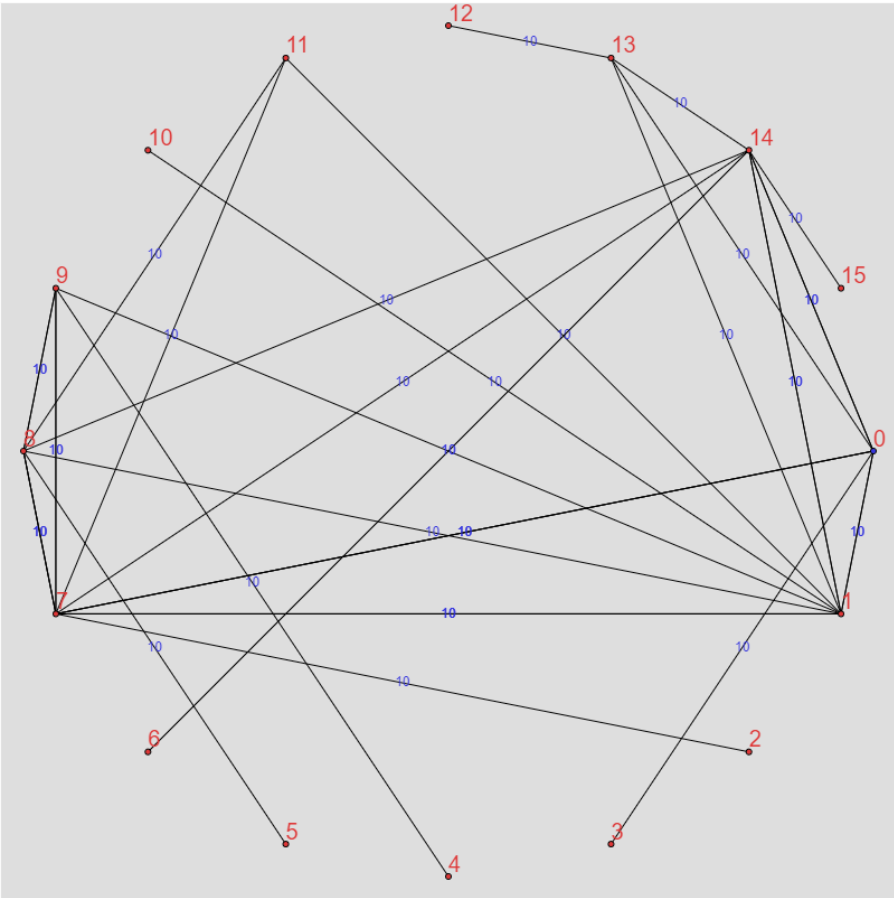
I tried to change the demand file to see what was going wrong.

In the end the demand file looked like this:

```
DEMANDS 1
label src dest bw
demand_0 0 1 120482
```

Which is only one demand of bandwidth 120482 from node 0 to node 1. (the number 240 is never read so I did not bother changing it)

The Airnet graph is the following :



As we can see, the weights (in blue) are always equal to 10. So in the OSPF case, the demand would go over the edge 0-1.

I copy the graph file of the topology below.

```
NODES 16
label x y
0_Los_Angeles -118.24368 34.05223
1_Chennai 80.27847 13.08784
2_NYIX 0.0 0.0
3_LAIX 0.0 0.0
4_NiXI 0.0 0.0
5_LINX 0.0 0.0
6_EQUINOX_IX 0.0 0.0
7_New_York -74.00597 40.71427
8_London -0.12574 51.50853
9_Mumbai 72.84794 19.01441
10_Palermo 13.35976 38.11582
11_Marseille 5.4 43.3
12_HKIX 0.0 0.0
13_Hong_Kong 114.15769 22.28552
14_Singapore 103.85007 1.28967
15_STIX 0.0 0.0

EDGES 74
label src dest weight bw delay
edge_0 0 1 10 1000000 24018
edge_1 1 0 10 1000000 24018
edge_2 0 1 10 1000000 24018
edge_3 1 0 10 1000000 24018
edge_4 0 3 10 1000000 13712
edge_5 3 0 10 1000000 13712
edge_6 0 13 10 1000000 19411
edge_7 13 0 10 1000000 19411
edge_8 0 14 10 1000000 23526
edge_9 14 0 10 1000000 23526
edge_10 0 14 10 1000000 23526
edge_11 14 0 10 1000000 23526
edge_12 0 7 10 1000000 6559
edge_13 7 0 10 1000000 6559
edge_14 0 7 10 1000000 6559
edge_15 7 0 10 1000000 6559
edge_16 0 7 10 1000000 6559
edge_17 7 0 10 1000000 6559
edge_18 1 7 10 1000000 22442
edge_19 7 1 10 1000000 22442
edge_20 1 7 10 1000000 22442
edge_21 7 1 10 1000000 22442
edge_22 1 7 10 1000000 22442
edge_23 7 1 10 1000000 22442
edge_24 1 8 10 1000000 13679
edge_25 8 1 10 1000000 13679
edge_26 1 9 10 1000000 1722
edge_27 9 1 10 1000000 1722
edge_28 1 10 10 1000000 11835
edge_29 10 1 10 1000000 11835
edge_30 1 11 10 1000000 12988
edge_31 11 1 10 1000000 12988
edge_32 1 13 10 1000000 6204
edge_33 13 1 10 1000000 6204
edge_34 1 14 10 1000000 4848
edge_35 14 1 10 1000000 4848
edge_36 1 14 10 1000000 4848
edge_37 14 1 10 1000000 4848
edge_38 2 7 10 1000000 13712
edge_39 7 2 10 1000000 13712
edge_40 4 9 10 1000000 13712
```

```
edge_41 9 4 10 1000000 13712
edge_42 5 8 10 1000000 13712
edge_43 8 5 10 1000000 13712
edge_44 6 14 10 1000000 13712
edge_45 14 6 10 1000000 13712
edge_46 7 8 10 1000000 9281
edge_47 8 7 10 1000000 9281
edge_48 7 8 10 1000000 9281
edge_49 8 7 10 1000000 9281
edge_50 7 8 10 1000000 9281
edge_51 8 7 10 1000000 9281
edge_52 7 9 10 1000000 20893
edge_53 9 7 10 1000000 20893
edge_54 7 9 10 1000000 20893
edge_55 9 7 10 1000000 20893
edge_56 7 11 10 1000000 10508
edge_57 11 7 10 1000000 10508
edge_58 7 14 10 1000000 25551
edge_59 14 7 10 1000000 25551
edge_60 8 9 10 1000000 11987
edge_61 9 8 10 1000000 11987
edge_62 8 9 10 1000000 11987
edge_63 9 8 10 1000000 11987
edge_64 8 11 10 1000000 1674
edge_65 11 8 10 1000000 1674
edge_66 8 14 10 1000000 18083
edge_67 14 8 10 1000000 18083
edge_68 12 13 10 1000000 13712
edge_69 13 12 10 1000000 13712
edge_70 13 14 10 1000000 4313
edge_71 14 13 10 1000000 4313
edge_72 14 15 10 1000000 13712
edge_73 15 14 10 1000000 13712
```

We can now also see that all bandwidths are equal to 1000000.

The maximum link utilisation for our simple test file should therefore be, if OSPF is used, equal to 120482/1000000 = 0.120482

This is the result I obtain even after optimisation with 2-SR.

On the other hand, the Repetita implementation says that the pre-optimisation maximum link utilisation is 0.060241 and post-optimisation is 0.040160666 as shown below.

```
[dragon1-h0:~/dev/Repetita :]$./repetita -graph ./data/2016TopologyZooUCL_inverseCapacity/Airtel.graph -demands data/2016Topolog
57266405376
Academic license - for non-commercial use only - expires 2022-11-04
pre-optimization max link utilization 0.060241
post-optimization max link utilization 0.040160666666666664
1 demands over 1 (100.0%) have a different segment routing path between pre-optimization and post-optimization
```

The value 0.060241 does not come from nowhere, it is equal to 0.120482/2. The value 0.040160666 then comes from 0.120482/3.

Still I do not get how they arrive at these values. It seems to be an error in my opinion, especially as the SR-path chosen is 0 → 7 → 1 (found from the outpaths file).

```
***Next hops priority 2 (sr paths)***

Destination 1_Chennai
sequence of middlepoints: 0_Los_Angeles -> 7_New_York -> 1_Chennai

***Next hops priority 3 (ecmp paths)***
```

And knowing that there exist direct links from 0 → 7 and 7 → 1, both edges having the same bandwith which is the same as any other edge in the graph. The maximum utilisation should not have changed.