Homework 3: STAGE

Results

This code was excecuted using Jupyter Notebook.

Traffic Data	Hyperparameters	Traffic weight zero-Load latency before STAGE (Mesh Network)	Traffic weighted zero-load latency after STAGE	Percentage Reduction in latency
Random	CountRepeat=50,nIterations=100	42060.094	28986.125	31.084
Uniform	CountRepeat=50,nIterations=100	33315.391	22857.933	31.389
Complement	CountRepeat=50,nIterations=100	9726.589	6467.449	33.507

Simulated Annealing and STAGE comparison

Traffic weighted Zero-Load latency

Traffic Data	Traffic weighted zero-load latency after performing SA	Traffic weight zero-Load latency after performing STAGE
Random	30846.946	28986.125
Uniform	24171.568	22857.933
Complement	6686.5813	6467.449

Table 1

Runtime

Traffic Data	Total execution time for SA in seconds	Total execution time for STAGE in seconds
Random	5056.870	6084.595
Uniform	5045.688	6031.259
Complement	5117.317	6002.592

Table 2

Table 1 gives us an idea about how well both the algorithms i.e. STAGE and Simulated Annealing has performed. Hence, we can infer that, for each traffic pattern that was given, STAGE algorithm performs better than SA algorithm. In other words the quality of the result is better for STAGE when compared to Simulated Annealing. STAGE gives around 4% more reduction in Traffic Weighted zero load latency than Simulated Annealing for each traffic pattern.

Table 2 tells us about the total time taken to run the respective algorithms and we can see that STAGE algorithm takes longer time to run than Simulated Annealing. Stage takes roughly around 15% more time than Simulated Annealing to run. This can be attributed to the additional computation overhead of calculating the function approximator on V^{π}.

Now, we are left with a classic trade-off between Runtime and Quality. STAGE has learned to optimize by improving on the performance of Hill-climbing by building a secondary evaluation function for search making STAGE a simple yet Robust predictive evaluation function.

Adjacency matrices of all the three configurations are given as separate excel files with the submission.

Output for Complement Traffic:

```
Tile placement vector:
[16, 24, 6, 48, 5, 21, 56, 23, 8, 40, 36, 20, 60, 32, 13, 52, 44, 29, 1
0, 33, 47, 51, 34, 18, 63, 59, 50, 28, 25, 38, 35, 0, 46, 31, 39, 61, 1
, 42, 45, 26, 62, 12, 3, 17, 55, 2, 14, 22, 19, 58, 54, 7, 30, 11, 9, 1
5, 41, 57, 53, 37, 43, 4, 27, 49]
```

Output for uniform traffic dataset:

```
Tile placement vector:
 [4, 2, 25, 1, 39, 21, 33, 60, 26, 55, 15, 50, 52, 30, 63, 9, 48, 38, 11
, 24, 8, 47, 43, 10, 34, 29, 12, 42, 32, 6, 27, 37, 41, 59, 13, 61, 44,
46, 0, 36, 17, 35, 53, 56, 49, 28, 31, 57, 7, 19, 23, 5, 62, 16, 45, 18
, 14, 40, 51, 20, 54, 22, 3, 58]
```

Output for Random traffic dataset:

```
Tile placement vector:
[47, 9, 56, 19, 16, 63, 62, 37, 48, 20, 44, 40, 30, 23, 2, 43, 42, 52,
31, 35, 54, 7, 39, 14, 6, 18, 15, 11, 34, 26, 25, 29, 60, 41, 59, 5, 13
, 27, 45, 12, 57, 58, 8, 50, 28, 0, 10, 55, 38, 53, 3, 21, 33, 32, 22,
51, 49, 1, 4, 17, 24, 36, 46, 61]
```