Homework 2: Simulated Annealing

Results

This code was excecuted using Jupyter Notebook.

Hyperparameter selection

From different combinations of the Hyperparameters (*T*, *Tth* ,*α* ,*Num_iter*) the following conclusions are made:

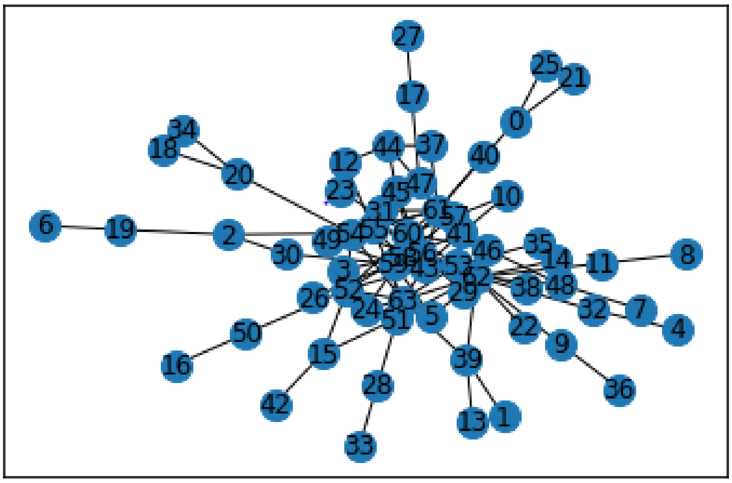
- As we increase the Number of iterations (*Num_iter*) the time taken to execute the whole program increases.
- If we decrease *α* (decay rate) the time taken to run the program decrease but, the algorithm may not find the global minima of the cost function
- By increasing the *α* (decay rate) value the SA algorithm searches the design space thoroughly to get the min value.
- After trying a lot of combinations, one combination gave best results for me, which is for T=100, Tth=0.1, α=0.9, Num_Iter=100. Hence, this combination of hyperparmeters has been used to run the SA algorithm for all the three traffic patterns.

Challenges:

It took a lot of time to understand the question. Once I understood the problem, implementing the Perturbation function for the mesh was really hard most of the time the network was getting disconnected for me. Finally, the SA algorithm took a lot of time to run completely hence it was difficult for me to debug.

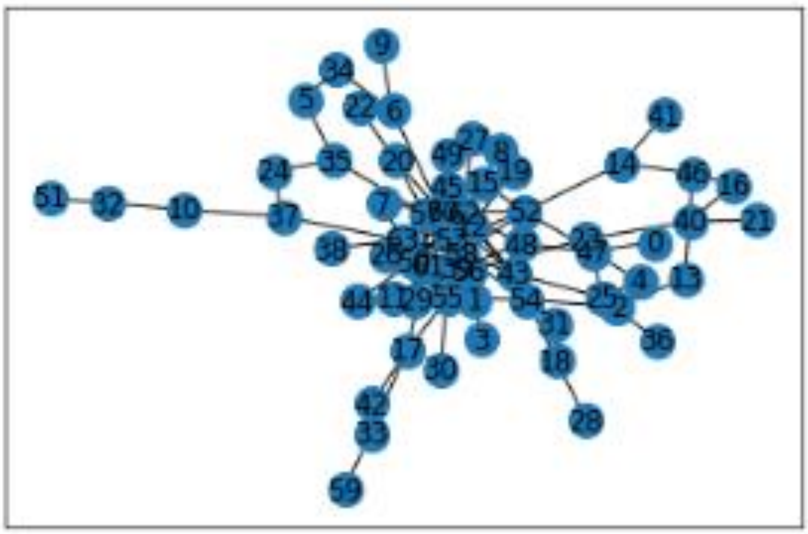
Output for Complement Traffic:

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



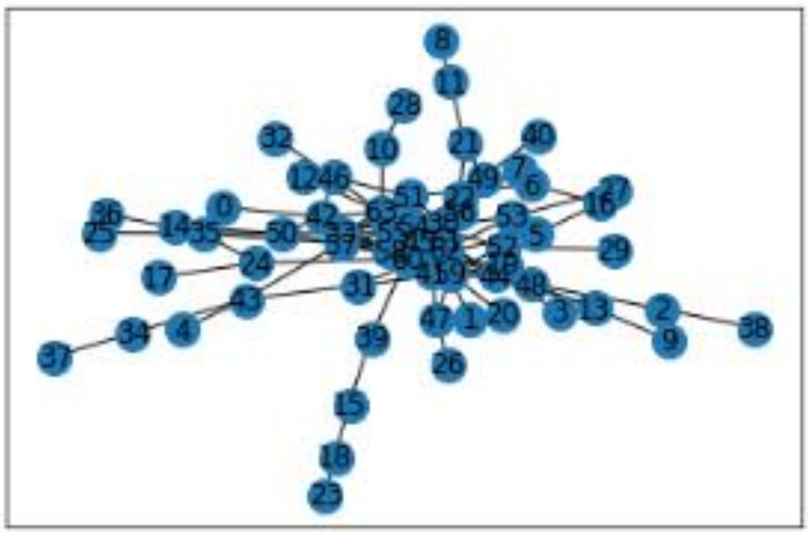
Output for uniform traffic dataset:

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



Output for Random traffic dataset:

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



Traffic Data	Hyperparameters	Traffic weighted zero-load latency after SA	Traffic weight zero-Load latency before SA
Random	T=500,Tth=0.1,α=0.9,nIterations=100	30846.946	42060.094
Uniform	T=100,Tth=0.1,α=0.9,nIterations=100	24171.568	33315.391
Complement	T=100,Tth=0.1,α=0.9,nIterations=100	6686.5813	9726.589

For Complement Traffic, zero-load latency for mesh configuration before annealing is 9726.5898 and it got reduced to 6686.5813 after the implementation of SA algorithm. We are able to reduce up to 31% reduction in latency.
For Random Traffic, zero-load latency for mesh configuration before annealing is 43422.867 and it got reduced to 30846.946 after the implementation of SA algorithm. We are able to achieve up to 27% reduction in latency.
For Uniform Traffic, zero-load latency for mesh configuration before annealing is 34384.452 and it got reduced to 24171.568 after the implementation of SA algorithm. We are able to achieve up to 26% reduction in latency.
We can see that the Traffic weighted zero-load latency is highest for random traffic dataset than all the other datasets and Traffic weighted zero-load latency is least for Complement traffic dataset

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