PYTHON QUESTIONS – REVIEW AND ANALYSIS REPORT
(Submitted by Anirudh M. and Tushar B.)

Question 1: Algorithm performance

Download the Sioux Falls network and trips file from Transportation Network Test Problems. You should be able to solve for user equilibrium using your code from Homework 4 with the following Python commands:

```
from network import *
net = Network("SiouxFalls_net.tntp", "SiouxFalls_trips.tntp")
net.userEquilibrium("FW", 20, 1e-4, net.averageExcessCost)
```

See the documentation of `userEquilibrium` from the previous assignment to see what the arguments do. After running this code, the link flows and costs can be seen in the `flow` and `cost` attributes of `net.link`.

For the first question on the assignment, compare the performance of MSA and Frank-Wolfe on this network. What is the gap level achieved for different numbers of iterations? What is the gap level achieved for different amounts of computational time? Experiment with different choices of step size for MSA (e.g., constant step size, different sequences) and different parameters in FW (e.g., different precision for solving for the step size, or using Newton's method instead of bisection). Describe your experiments, results, and conclusions, using tables and/or figures to illustrate your findings. What seems to be fastest?

Solution:

The gap values **(Relative Gap and AEC)** found after running Franke-Wolfe and MSA on the network for different iterations chosen (20, 100, 1000, 10000) are summarized in the table below:

Franke-Wolfe Method (Tolerance Level = $1e^{-4}$)		
No. of Iterations	**Relative Gap**	**AEC**
20	0.012141	0.251064
100	0.001446	0.029956
1000	-	0.002566
10000	-	0.000394

Table 1.1

For the Franke-Wolfe method, the tolerance level of $1e^{-4}$ is achieved in **Iteration 986** with a Relative Gap value of 0.00009.

MSA Method (Tolerance Level = $1e^{-4}$)		
No. of Iterations	**Relative Gap**	**AEC**
20	0.053370	1.178939
100	0.009846	0.206518
1000	0.001000	0.020769
10000	-	0.002050

Table 1.2

For the MSA method, the tolerance level of $1e^{-4}$ is achieved in **Iteration 9757** with a Relative Gap value of 0.000099.

Comparison between Frank-Wolfe and MSA:

The link flows for the 10000th iteration and the associated costs for both the methods are reported in the table below (using AEC as the gap value):

Nodes		**Franke-Wolfe**		**MSA**	
Tail Node	**Head Node**	**Link Flows**	**Associated Costs**	**Link Flows**	**Associated Costs**
1	2	4496.740326	6.000817751	4494.587571	6.000816186
1	3	8121.647835	4.00870175	8118.964462	4.008690256
2	1	4521.227877	6.00083571	4518.987871	6.000834055
2	6	5967.963204	6.574259521	5967.164547	6.573416995
3	1	8097.160284	4.008597277	8094.564162	4.008586257
3	4	14012.12879	4.26984509	14008.31554	4.269551469
3	12	10030.13699	4.020242188	10023.83892	4.020191394
4	3	14036.22638	4.271706167	14032.18666	4.271393506
4	5	18010.44896	2.315659897	18007.7281	2.315469192
4	11	5202.689731	7.135647126	5201.201308	7.134348107
5	4	18034.77652	2.317368863	18031.77966	2.317157964
5	6	8798.170183	9.997959244	8798.188315	9.998008688
5	9	15783.0017	9.653928176	15781.51469	9.652174523
6	2	5992.450755	6.600256782	5991.564846	6.599310681
6	5	8806.029397	10.01941936	8806.463276	10.02060577

6	8	12493.9506	14.6951215	12492.74529	14.69022334
7	8	12102.81972	5.55324928	12101.51037	5.552144559
7	18	15795.78042	2.062253583	15795.93298	2.062255988
8	6	12526.29737	14.82710333	12525.42056	14.82351222
8	7	12041.94581	5.502266921	12042.29166	5.502554404
8	9	6883.581642	15.17746502	6882.648914	15.1746594
8	16	8388.841116	10.72982337	8387.280521	10.72556083
9	5	15799.47005	9.673382669	15797.29128	9.670805328
9	8	6837.276272	15.03955069	6836.842041	15.03827058
9	10	21745.54545	5.683258218	21744.67127	5.682826776
10	9	21815.70843	5.7180568	21814.64099	5.717524863
10	11	17727.47725	12.40711369	17726.68726	12.40579344
10	15	23127.37726	13.72448089	23126.08604	13.72275598
10	16	11047.28527	20.08592468	11046.94332	20.0839331
10	17	8100.589941	16.31043778	8100.058179	16.30825584
11	4	5302.459754	7.225296581	5301.020876	7.223967134
11	10	17605.47945	12.20531033	17604.08399	12.20302614
11	12	8366.083392	13.59312335	8365.360596	13.59049963
11	14	9776.752325	13.69379514	9776.436279	13.69254174
12	3	9981.551845	4.019852823	9975.567505	4.019805255
12	11	8405.79942	13.73833967	8404.725376	13.73438538
12	13	12292.67073	3.022834158	12289.17747	3.022808213
13	12	12383.80162	3.023518843	12380.27083	3.023492032
13	24	11121.52785	17.66184247	11121.16857	17.66007717
14	11	9814.808519	13.84561182	9814.287797	13.84352256
14	15	9036.616338	12.23524288	9036.472385	12.23478186
14	23	8400.089991	9.078505495	8399.986342	9.078254845
15	10	23193.42066	13.81309281	23192.38387	13.81169587
15	14	9079.882012	12.37480529	9079.696727	12.37420335
15	19	19084.48155	4.326543408	19083.33644	4.326225056
15	22	18409.74903	9.087897635	18409.36581	9.087390757
16	8	8406.619338	10.77855018	8406.543949	10.7783429
16	10	11073.39266	20.23852418	11073.17659	20.23725677
16	17	11694.52055	9.500220956	11693.97347	9.498817591
16	18	15285.0315	3.163751998	15279.27352	3.163505391
17	10	8100.59994	16.31047881	8100.100068	16.30842771
17	16	11683.34167	9.471583975	11683.07624	9.470905038
17	19	9953.244676	7.437114584	9952.783492	7.436106938
18	7	15856.65433	2.063218798	15855.15169	2.063194838
18	16	15340.09599	3.166124453	15335.66745	3.165932702
18	20	18981.38186	4.259621886	18978.46047	4.259462091
19	15	19117.69823	4.335802981	19116.49774	4.335467487
19	17	9942.075792	7.412750927	9941.928154	7.412429421
19	20	8688.291171	9.458872829	8687.892347	9.45787057
20	18	18997.32027	4.260494989	18994.0731	4.260316932
20	19	8710.338966	9.514494822	8710.198303	9.514138618
20	21	6303.829617	8.168145396	6302.326581	8.166078312

20	22	7001.719828	7.715797336	7000.283096	7.713568928
21	20	6241.825817	8.084093112	6240.31872	8.082081013
21	22	8620.877912	4.214880765	8619.343074	4.21330386
21	24	10309.41906	11.92408841	10309.28618	11.92362831
22	15	18385.84142	9.05633527	18385.72669	9.05618411
22	20	7001.709829	7.715781822	7000.209543	7.713454882
22	21	8608.609139	4.202299254	8607.30905	4.200969173
22	23	9662.593741	12.36847097	9661.691203	12.36534477
23	14	8394.880512	9.065919074	8394.613519	9.065274633
23	22	9626.407359	12.24381384	9625.944503	12.24222843
23	24	7905.049495	3.761144196	7902.783517	3.75912574
24	13	11112.65873	17.6183148	11112.26193	17.61636978
24	21	10259.68403	11.75312308	10259.31234	11.7518547
24	23	7863.653635	3.724543075	7861.663994	3.72279838

Table 1.3

<u>Comparison for MSA using different step sizes:</u>

The step size for the above MSA implementation uses the sequence $\frac{1}{i+1}$ where 'i' is the i^{th} iteration. Next, we experimented with taking a constant step size of 0.01 and ran the algorithm for 10000 iterations. Following this, we tried out another step size with a different sequence this time to see the results. The step size we used was $\frac{1}{(i+1)^{\left(\frac{2}{3}\right)}}$ where 'i' is the i^{th} iteration. The results are summarized in the table below:

AEC after 10000 iterations (Step Size = 1/(1+i)): 0.002050
AEC after 10000 iterations (Step Size = $1/(1+i)^{2/3}$): 0.002577
AEC after 10000 iterations (Step Size = 0.01): 0.020729

Nodes		**Step Size = 0.01**		**Step Size = $1/(1+i^{2/3})$**	
Tail Node	**Head Node**	**Link Flows**	**Associated Costs**	**Link Flows**	**Associated Costs**
1	2	4498.991	6.000819	4496.117	6.000817
1	3	8118.362	4.008688	8122.427	4.008705
2	1	4518.362	6.000834	4522.427	6.000837
2	6	5954.139	6.559723	5965.975	6.572162
3	1	8098.991	4.008605	8096.117	4.008593
3	4	14092.77	4.276111	14019.55	4.270418
3	12	10052.58	4.020424	10038.09	4.020306
4	3	14079.07	4.275039	14045.81	4.272449
4	5	18092.77	2.321471	18019.55	2.316299
4	11	5200	7.1333	5200	7.1333
5	4	18079.07	2.320498	18045.81	2.318146
5	6	8776.724	9.939691	8800.78	10.00508
5	9	15871.19	9.758823	15788.63	9.660572

6	2	5973.51	6.58012	5992.284	6.600079
6	5	8785.494	9.963466	8799.367	10.00122
6	8	12475.72	14.62117	12496.9	14.7071
7	8	12152.29	5.595256	12103.16	5.55354
7	18	15910.97	2.06409	15824.91	2.062714
8	6	12503.86	14.73543	12521.79	14.80867
8	7	12100.66	5.551431	12048.2	5.507471
8	9	6910.793	15.25982	6884.016	15.17877
8	16	8357.372	10.64433	8376.947	10.6974
9	5	15848.72	9.731933	15816.3	9.693324
9	8	6882.322	15.17368	6828.252	15.013
9	10	21788.87	5.704707	21760.38	5.690588
10	9	21837.93	5.729149	21832.28	5.726326
10	11	17752.62	12.44923	17711.79	12.38093
10	15	23127.56	13.72473	23140.92	13.7426
10	16	11034.69	20.01267	11043.86	20.06601
10	17	8100	16.30802	8100	16.30802
11	4	5300	7.223025	5300	7.223025
11	10	17595.85	12.18957	17610.47	12.21349
11	12	8360.676	13.57351	8355.651	13.55532
11	14	9791.969	13.75428	9775.479	13.68874
12	3	10046.91	4.020378	9985.528	4.019884
12	11	8365.347	13.59045	8399.887	13.71659
12	13	12313.26	3.022987	12293.74	3.022842
13	12	12412.25	3.023736	12385.42	3.023531
13	24	11087.75	17.4966	11114.58	17.62776
14	11	9830.531	13.90885	9829.924	13.9064
14	15	9051.618	12.28341	9023.733	12.19407
14	23	8363.217	8.98992	8403.414	9.086548
15	10	23209.44	13.8347	23188.92	13.80703
15	14	9059.456	12.30867	9090.808	12.41037
15	19	19070.28	4.322599	19086.45	4.32709
15	22	18407.39	9.084773	18397.58	9.071815
16	8	8362.353	10.6578	8402.646	10.76763
16	10	11058.64	20.15214	11069.09	20.21331
16	17	11651.65	9.390849	11684.43	9.47437
16	18	15399.78	3.168725	15298.71	3.164339
17	10	8100	16.30802	8100	16.30802
17	16	11633.76	9.345562	11673.75	9.447077
17	19	9955.55	7.442154	9950.171	7.430402
18	7	15962.6	2.064925	15879.87	2.06359
18	16	15446.6	3.170786	15360.32	3.167002
18	20	19047.48	4.263257	18995.55	4.260398
19	15	19110.57	4.333813	19119.83	4.3364
19	17	9937.66	7.403141	9939.49	7.407122
19	20	8679.346	9.436425	8685.797	9.452607
20	18	19045.93	4.263172	19012.12	4.261308

20	19	8701.75	9.492776	8708.504	9.50985
20	21	6316.14	8.185131	6305.265	8.17012
20	22	7000	7.71313	7000	7.71313
21	20	6236.996	8.077651	6244.542	8.087723
21	22	8556.555	4.149514	8604.751	4.198354
21	24	10266.09	11.77499	10303.79	11.9046
22	15	18356.81	9.018171	18379.26	9.047666
22	20	7000	7.71313	7000	7.71313
22	21	8549.946	4.14288	8598.522	4.191995
22	23	9669.256	12.39157	9655.523	12.344
23	14	8393.941	9.063651	8390.784	9.056039
23	22	9612.067	12.1948	9630.974	12.25947
23	24	7920.658	3.775095	7902.47	3.758847
24	13	11086.74	17.49172	11106.26	17.58696
24	21	10193.55	11.52961	10249.29	11.71772
24	23	7894.193	3.75149	7865.292	3.72598

Table 1.4

Comparison for Frank-Wolfe using different Precision Values:

The final part of the question asks us to experiment with different parameters for the Franke-Wolfe algorithm by using both different precision values for calculating the step size. We had already implemented Franke-Wolfe by calculating the step size using 10^{-4} precision and the results can be found in Tables 1.2 and 1.3.

AEC after 10000 iterations (With 10^{-4} Precision): 0.000394
AEC after 10000 iterations (With 10^{-5} Precision): 0.000285
AEC after 10000 iterations (With 10^{-3} Precision): 0.001922

Nodes		With 10^{-5} Precision		With 10^{-3} Precision	
Tail Node	**Head Node**	**Link Flows**	**Associated Costs**	**Link Flows**	**Associated Costs**
1	2	4494.80698	6.00081635	4496.26458	6.00081741
1	3	8119.26269	4.00869153	8121.11161	4.00869945
2	1	4519.28536	6.00083427	4521.11166	6.00083562
2	6	5967.34469	6.57360701	5966.60657	6.57282857
3	1	8094.7843	4.00858719	8096.26453	4.00859347
3	4	14007.8026	4.26951199	14011.5742	4.26980237
3	12	10023.6692	4.02019003	10026.1391	4.02020993
4	3	14031.7124	4.27135681	14034.6853	4.27158686
4	5	18007.7779	2.31547268	18011.5735	2.31573874
4	11	5201.00559	7.13417737	5200.00166	7.13330193
5	4	18031.7838	2.31715826	18034.6849	2.31736241
5	6	8797.92014	9.99727744	8794.88198	9.98899763
5	9	15781.9024	9.65263171	15787.0325	9.65868426
6	2	5991.82308	6.59958641	5991.45365	6.59919196

6	5	8806.01428	10.019378	8802.65967	10.010211
6	8	12492.6631	14.6898895	12491.1465	14.6837284
7	8	12101.8876	5.55246277	12101.1877	5.5518724
7	18	15794.7922	2.06223801	15805.4897	2.06240679
8	6	12525.2357	14.822755	12523.7713	14.8167595
8	7	12040.9876	5.50147061	12041.0591	5.50152995
8	9	6882.97091	15.1756278	6883.4957	15.1772065
8	16	8388.43224	10.7287064	8387.37265	10.7258124
9	5	15797.8142	9.67142386	15802.3662	9.67681025
9	8	6836.77149	15.0380626	6837.28517	15.0395769
9	10	21744.6892	5.6828356	21749.747	5.68533261
10	9	21814.4016	5.71740557	21818.8702	5.71963284
10	11	17726.4087	12.4053279	17728.8485	12.4094057
10	15	23126.3184	13.7230664	23123.1513	13.7188366
10	16	11047.0994	20.0848421	11048.0974	20.0906554
10	17	8100.05762	16.3082535	8100.0001	16.3080175
11	4	5300.90934	7.22386412	5300.00137	7.22302582
11	10	17603.9288	12.2027721	17602.2122	12.1999631
11	12	8365.33488	13.5904063	8364.94732	13.5889997
11	14	9776.23107	13.691728	9782.03466	13.7147622
12	3	9975.2811	4.01980298	9978.18093	4.01982602
12	11	8404.85228	13.7348525	8403.53607	13.7300085
12	13	12288.9685	3.02280666	12291.0867	3.02282239
13	12	12380.0978	3.02349072	12381.7173	3.02350301
13	24	11121.1349	17.6599117	11118.2853	17.6459168
14	11	9814.13754	13.8429198	9816.8093	13.8536425
14	15	9036.08065	12.2335274	9040.93781	12.2490929
14	23	8400.43639	9.07934325	8398.95187	9.07575372
15	10	23192.697	13.8121178	23192.813	13.812274
15	14	9079.97804	12.3751173	9078.98111	12.3718788
15	19	19083.2312	4.3261958	19080.776	4.32551344
15	22	18408.8603	9.0867222	18406.1474	9.08313498
16	8	8406.30426	10.7776839	8406.07927	10.7770654
16	10	11072.8716	20.2354681	11074.1951	20.2432316
16	17	11694.1727	9.4993286	11688.8862	9.48577728
16	18	15279.8724	3.16353103	15288.5279	3.16390188
17	10	8100.09906	16.3084236	8100.00011	16.3080176
17	16	11683.0773	9.47090778	11677.8045	9.45742984
17	19	9952.9972	7.43657386	9953.62051	7.43793586
18	7	15855.6921	2.06320345	15865.6184	2.06336187
18	16	15334.612	3.16588703	15344.414	3.16631158
18	20	18978.3265	4.25945476	18987.3407	4.25994805
19	15	19116.5209	4.33547396	19114.7224	4.33497147
19	17	9941.94327	7.41246234	9942.5388	7.41375929
19	20	8688.07016	9.45831741	8685.04973	9.45073097
20	18	18993.966	4.26031106	19003.3555	4.26082617
20	19	8710.30592	9.51441113	8707.91442	9.50835749

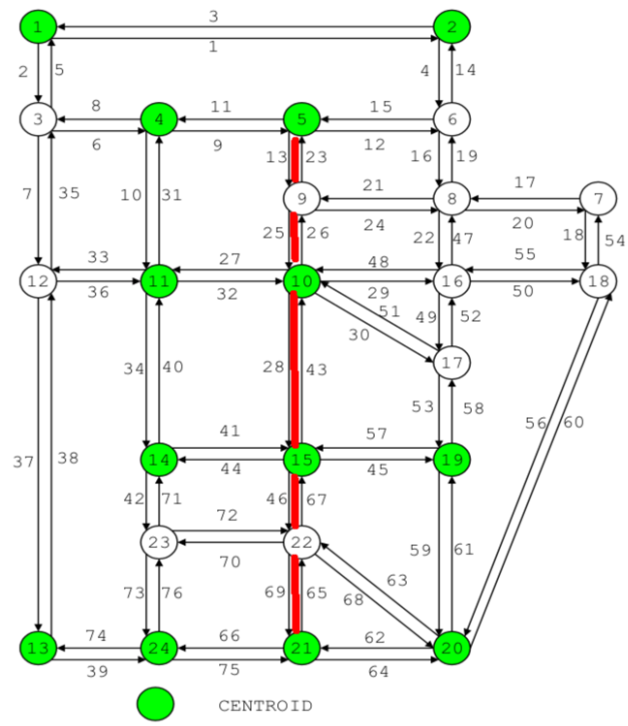
20	21	6302.43613	8.16622892	6303.81568	8.16812622
20	22	7000.29087	7.71358099	7000.00044	7.71313068
21	20	6240.48996	8.08230955	6242.69521	8.08525448
21	22	8619.16086	4.21311671	8615.95191	4.20982273
21	24	10309.1357	11.9231072	10305.3618	11.9100485
22	15	18385.8467	9.05634218	18379.9061	9.04851857
22	20	7000.11235	7.71330418	7000.00032	7.7131305
22	21	8607.14422	4.20080059	8601.54698	4.19508142
22	23	9661.5	12.3646826	9660.67123	12.3618128
23	14	8394.44547	9.06486904	8395.68321	9.06785691
23	22	9626.29116	12.2434158	9620.02482	12.2219722
23	24	7902.94771	3.75927194	7903.55329	3.75981124
24	13	11112.2642	17.6163808	11108.9159	17.5999772
24	21	10259.2061	11.7514923	10258.6463	11.7495821
24	23	7861.74795	3.72287198	7859.63822	3.72102336

Table 1.5

Although the AEC is lower in the case of 10^{-5} precision, the time taken for the algorithm to run is 211.48 seconds whereas for a precision value of 10^{-4}, it is 197.50 seconds. So, the decision to choose the value of the precision depends upon the needs of the user. If algorithm run time is a priority, a precision of 10^{-4} seems the more likely choice but if faster convergence is needed, 10^{-5} can be chosen as the precision value.

Question 2: Effects of a network change

In this question, you will evaluate the impact of building a new highway in Sioux Falls. Here is a schematic of the Sioux Falls network, based on a diagram created by Hai Yang and Meng Qiang from the Hong Kong University of Science and Technology. The numbers indicate the order of the links in the network file. The red line indicates where the new highway is to be built.



Assume that this highway will double the capacity of each of the highlighted links (in both directions) and reduce the free-flow time by half. Your task is to forecast the effects of bulding this highway. Use equilibrium traffic assignment, and make sure you solve to a small enough gap that the link flows are stable. What will be the change in total system travel time? What links will see increased flow, and what links will see decreased flow? What links will see higher travel times, and what links will see lower travel times? Is there anybody who will have a higher travel time afterwards? Who are the "winners" and "losers" if this highway were to be built?

Document your analysis and conclusions, using tables and/or figures to illustrate your findings.

Solution:

After making edits to the input file to accommodate the changes made in the network, we ran the Franke-Wolfe algorithm for 10000 iterations using the Bisection method to calculate the step size and obtained the new link flows and associated costs.

Nodes		With No Network Changes		With Network Changes	
Tail Node	**Head Node**	**Link Flows**	**Associated Costs**	**Link Flows**	**Associated Costs**
1	2	4494.588	6.000816	4266.006	6.000662
1	3	8118.964	4.00869	7874.133	4.007688
2	1	4518.988	6.000834	4273.977	6.000667
2	6	5967.165	6.573417	6065.994	6.680273
3	1	8094.564	4.008586	7866.162	4.007657
3	4	14008.32	4.269551	17304.93	4.627737
3	12	10023.84	4.020191	13073.66	4.058428
4	3	14032.19	4.271394	17337.99	4.632547
4	5	18007.73	2.315469	22370.61	2.751333
4	11	5201.201	7.134348	5134.298	7.0771
5	4	18031.78	2.317158	22534	2.773525
5	6	8798.188	9.998009	8533.979	9.309338
5	9	15781.51	9.652175	22136.79	3.06282
6	2	5991.565	6.599311	6073.965	6.689122
6	5	8806.463	10.02061	8526.213	9.290037
6	8	12492.75	14.69022	10599.97	8.57744
7	8	12101.51	5.552145	10664.47	4.539232
7	18	15795.93	2.062256	10435.61	2.01186
8	6	12525.42	14.82351	10600.18	8.577948
8	7	12042.29	5.502554	10632.53	4.520875
8	9	6882.649	15.17466	7064.147	15.74246
8	16	8387.281	10.72556	6100.243	6.602219
9	5	15797.29	9.670805	22307.95	3.08043
9	8	6836.842	15.03827	7032.594	15.64055
9	10	21744.67	5.682827	34400.75	2.025174
10	9	21814.64	5.717525	34640.36	2.039959
10	11	17726.69	12.40579	15156.89	8.958231
10	15	23126.09	13.72276	32834.7	3.980722
10	16	11046.94	20.08393	10501.57	17.1353
10	17	8100.058	16.30826	7947.481	15.69973
11	4	5301.021	7.223967	5103.957	7.051864
11	10	17604.08	12.20303	15079.85	8.878368
11	12	8365.361	13.5905	5692.212	7.627256
11	14	9776.436	13.69254	6563.527	5.969086
12	3	9975.568	4.019805	13032.63	4.057698
12	11	8404.725	13.73439	5708.848	7.646363
12	13	12289.18	3.022808	10803.57	3.013623
13	12	12380.27	3.023492	10879.18	3.014008

13	24	11121.17	17.66008	9020.28	9.911962
14	11	9814.288	13.84352	6539.513	5.940426
14	15	9036.472	12.23478	8499.749	10.6631
14	23	8399.986	9.078255	4860.627	4.569337
15	10	23192.38	13.8117	33015.43	4.002493
15	14	9079.697	12.3742	8499.908	10.66352
15	19	19083.34	4.326225	19370.28	4.407807
15	22	18409.37	9.087391	26584.61	2.32727
16	8	8406.544	10.77834	6100.06	6.602026
16	10	11073.18	20.23726	10522.85	17.24213
16	17	11693.97	9.498818	10867.72	7.593687
16	18	15279.27	3.163505	11633.49	3.054949
17	10	8100.1	16.30843	7962.11	15.75658
17	16	11683.08	9.470905	10843.58	7.544147
17	19	9952.783	7.436107	9905.844	7.334279
18	7	15855.15	2.063195	10467.55	2.012005
18	16	15335.67	3.165933	11678.74	3.055809
18	20	18978.46	4.259462	15644.78	4.119814
19	15	19116.5	4.335467	19417.23	4.421506
19	17	9941.928	7.412429	9896.331	7.313816
19	20	8687.892	9.457871	7874.217	7.682943
20	18	18994.07	4.260317	15621.97	4.119117
20	19	8710.198	9.514139	7911.655	7.753486
20	21	6302.327	8.166078	6542.295	8.515306
20	22	7000.283	7.713569	8091.219	9.84322
21	20	6240.319	8.082081	6489.976	8.435808
21	22	8619.343	4.213304	12727.53	1.32883
21	24	10309.29	11.92363	9735.279	10.09612
22	15	18385.73	9.056184	26618.55	2.331502
22	20	7000.21	7.713455	8058.162	9.764555
22	21	8607.309	4.200969	12726.75	1.328749
22	23	9661.691	12.36534	8799.989	9.757046
23	14	8394.614	9.065275	4836.454	4.558095
23	22	9625.945	12.24223	8800.089	9.757309
23	24	7902.784	3.759126	4361.035	2.16313
24	13	11112.26	17.61637	8995.893	9.848285
24	21	10259.31	11.75185	9683.739	9.947035
24	23	7861.664	3.722798	4336.962	2.159558

Table 2.1

Old TSTT: 7480162.915
New TSTT: 5108078.528

There is close to a 46% change in the Total System Travel Time. There are changes to the link flows and the ones that have changed by more than 20% have been captured.

Tail Node	Head Node	Old Link Flows	New Link Flows	Change %
1	2	4494.587571	4266.006173	-5%
1	3	8118.964462	7874.132732	-3%
2	1	4518.987871	4273.977091	-5%
2	6	5967.164547	6065.994384	2%
3	1	8094.564162	7866.161813	-3%
3	4	14008.31554	17304.93237	24%
3	12	10023.83892	13073.65505	30%
4	3	14032.18666	17337.98521	24%
4	5	18007.7281	22370.60511	24%
4	11	5201.201308	5134.298025	-1%
5	4	18031.77966	22533.99879	25%
5	6	8798.188315	8533.979285	-3%
5	9	15781.51469	22136.78937	40%
6	2	5991.564846	6073.965303	1%
6	5	8806.463276	8526.21279	-3%
6	8	12492.74529	10599.974	-15%
7	8	12101.51037	10664.47082	-12%
7	18	15795.93298	10435.6052	-34%
8	6	12525.42056	10600.17842	-15%
8	7	12042.29166	10632.53032	-12%
8	9	6882.648914	7064.146859	3%
8	16	8387.280521	6100.243126	-27%
9	5	15797.29128	22307.94954	41%
9	8	6836.842041	7032.594377	3%
9	10	21744.67127	34400.75056	58%
10	9	21814.64099	34640.35825	59%
10	11	17726.68726	15156.88515	-14%
10	15	23126.08604	32834.69634	42%
10	16	11046.94332	10501.56506	-5%
10	17	8100.058179	7947.481422	-2%
11	4	5301.020876	5103.95719	-4%
11	10	17604.08399	15079.84741	-14%
11	12	8365.360596	5692.212107	-32%

Tail Node	Head Node	Old Link Flows	New Link Flows	Change %
11	14	9776.436279	6563.527378	-33%
12	3	9975.567505	13032.63129	31%
12	11	8404.725376	5708.848013	-32%
12	13	12289.17747	10803.57256	-12%
13	12	12380.27083	10879.1847	-12%
13	24	11121.16857	9020.280493	-19%
14	11	9814.287797	6539.5129	-33%
14	15	9036.472385	8499.749057	-6%
14	23	8399.986342	4860.626794	-42%
15	10	23192.38387	33015.42525	42%
15	14	9079.696727	8499.907619	-6%
15	19	19083.33644	19370.27561	2%
15	22	18409.36581	26584.6143	44%
16	8	8406.543949	6100.059531	-27%
16	10	11073.17659	10522.85275	-5%
16	17	11693.97347	10867.72014	-7%
16	18	15279.27352	11633.49368	-24%
17	10	8100.100068	7962.110259	-2%
17	16	11683.07624	10843.5778	-7%
17	19	9952.783492	9905.844071	0%
18	7	15855.15169	10467.5457	-34%
18	16	15335.66745	11678.74012	-24%
18	20	18978.46047	15644.78196	-18%
19	15	19116.49774	19417.22705	2%
19	17	9941.928154	9896.330569	0%
19	20	8687.892347	7874.2167	-9%
20	18	18994.0731	15621.9689	-18%
20	19	8710.198303	7911.654638	-9%
20	21	6302.326581	6542.29455	4%
20	22	7000.283096	8091.218806	16%
21	20	6240.31872	6489.976379	4%
21	22	8619.343074	12727.52955	48%
21	24	10309.28618	9735.278861	-6%

Tail Node	Head Node	Old Link Flows	New Link Flows	Change %
22	15	18385.72669	26618.55032	45%
22	20	7000.209543	8058.161851	15%
22	21	8607.30905	12726.75086	48%
22	23	9661.691203	8799.988802	-9%
23	14	8394.613519	4836.453754	-42%
23	22	9625.944503	8800.08918	-9%
23	24	7902.783517	4361.034785	-45%
24	13	11112.26193	8995.892634	-19%
24	21	10259.31234	9683.739382	-6%
24	23	7861.663994	4336.962124	-45%

Table 2.2

Table 2.2 shows the new and old link flows along with the change percentage which has occurred due to the construction of the new highway. We can see that the links highlighted in green have a lesser travel time and the ones highlighted in red have had an increase in link travel time.

Tail Node	Head Node	Old Costs	New Costs	Change %
1	2	6.000816186	6.000662393	0%
1	3	4.008690256	4.007688488	0%
2	1	6.000834055	6.000667358	0%
2	6	6.573416995	6.6802728	2%
3	1	4.008586257	4.007657403	0%
3	4	4.269551469	4.627736909	8%
3	12	4.020191394	4.058427605	1%
4	3	4.271393506	4.632546639	8%
4	5	2.315469192	2.75133251	19%
4	11	7.134348107	7.077099922	-1%
5	4	2.317157964	2.773524941	20%
5	6	9.998008688	9.309338497	-7%
5	9	9.652174523	3.062820381	-68%
6	2	6.599310681	6.689121961	1%
6	5	10.02060577	9.290037444	-7%
6	8	14.69022334	8.577440428	-42%
7	8	5.552144559	4.539232422	-18%
7	18	2.062255988	2.011859614	-2%
8	6	14.82351222	8.577947833	-42%
8	7	5.502554404	4.52087486	-18%
8	9	15.1746594	15.74246267	4%
8	16	10.72556083	6.602219215	-38%
9	5	9.670805328	3.080430064	-68%
9	8	15.03827058	15.64055164	4%
9	10	5.682826776	2.025173665	-64%
10	9	5.717524863	2.039958978	-64%
10	11	12.40579344	8.958230699	-28%
10	15	13.72275598	3.98072174	-71%
10	16	20.0839331	17.13529602	-15%
10	17	16.30825584	15.69972819	-4%
11	4	7.223967134	7.051864481	-2%
11	10	12.20302614	8.878368332	-27%
11	12	13.59049963	7.627256432	-44%

Tail Node	Head Node	Old Costs	New Costs	Change %
11	14	13.692542	5.9690856	-56%
12	3	4.0198053	4.0576977	1%
12	11	13.734385	7.6463631	-44%
12	13	3.0228082	3.0136229	0%
13	12	3.023492	3.0140083	0%
13	24	17.660077	9.9119619	-44%
14	11	13.843523	5.9404256	-57%
14	15	12.234782	10.663099	-13%
14	23	9.0782548	4.5693368	-50%
15	10	13.811696	4.002493	-71%
15	14	12.374203	10.663522	-14%
15	19	4.3262251	4.4078073	2%
15	22	9.0873908	2.3272701	-74%
16	8	10.778343	6.6020263	-39%
16	10	20.237257	17.242126	-15%
16	17	9.4988176	7.593687	-20%
16	18	3.1635054	3.0549493	-3%
17	10	16.308428	15.756576	-3%
17	16	9.470905	7.5441475	-20%
17	19	7.4361069	7.3342788	-1%
18	7	2.0631948	2.0120055	-2%
18	16	3.1659327	3.0558091	-3%
18	20	4.2594621	4.1198141	-3%
19	15	4.3354675	4.4215065	2%
19	17	7.4124294	7.3138163	-1%
19	20	9.4578706	7.6829427	-19%
20	18	4.2603169	4.1191168	-3%
20	19	9.5141386	7.7534859	-19%
20	21	8.1660783	8.5153065	4%
20	22	7.7135689	9.8432203	28%
21	20	8.082081	8.4358077	4%
21	22	4.2133039	1.3288296	-68%
21	24	11.923628	10.096116	-15%

Tail Node	Head Node	Old Costs	New Costs	Change %
22	15	9.0561841	2.3315024	-74%
22	20	7.7134549	9.7645555	27%
22	21	4.2009692	1.3287491	-68%
22	23	12.365345	9.7570462	-21%
23	14	9.0652746	4.5580953	-50%
23	22	12.242228	9.7573088	-20%
23	24	3.7591257	2.1631303	-42%
24	13	17.61637	9.8482848	-44%
24	21	11.751855	9.9470346	-15%
24	23	3.7227984	2.1595582	-42%

Table 2.3

Table 2.3 clearly demonstrates the real winners and losers if the highway were to be built. The links highlighted in green have had a cost reduction if the highway is built and the ones in red are the losers where there is an increase in the costs associated. Looking at it from a bigger picture, we can see a lot more green links than red ones, and we can conclude that the highway will serve its purpose in reducing travel time for a majority of the travelers.

Question 3: Speeding up traffic assignment

The last question asks you to make changes to your code so that the run time for traffic assignment is reduced. You can make whatever changes you like to any parts of the code in any files. Your tests should involve networks of different size from the Transportation Networks Test Problems site. Document your changes, and their effects on the performance of the algorithm.

You will receive full credit for this problem as long as you clearly describe your experiments and their effectiveness, regardless of the amount of time you are able to save. However, there is an opportunity for extra credit. I will select a (fairly large) network and run each submission on this network on my machine. The individual or group whose code reaches a relative gap of 1e-6 fastest (as measured by wall clock time) will receive 10 bonus points on this assignment. The runner up will receive 5 bonus points.

Be sure to write your code so that I can run it using the three lines of code given at the start of Question 1 (replacing "Sioux Falls" with the network of my choice). I will independently verify that the flows returned by your code are feasible and correspond to a sufficiently small relative gap.

Solution:

We tried to implement multiple changes only successful in a couple of those:

1. Tried to vectorize the code but was not able to. It involved changing all the data structures and still faced some issues.
2. We were able to implement a version of Newton's method that works faster than the bisection method. It is not aggressively fast, but it is still better, decreasing time taken from 197.50 secs (Bisection) to 180.66 secs (Newton's). We developed new methods in the Links class to include the function for Cost Derivative and to Calculate Cost Derivative.
3. We observed that it took at least four iterations to get to Frank Wolfe Precision for Lambda for Newton's method to work, so tried to change that by multiplying lambda by a 0.2 term but that increased the time taken from 180.66 secs to 309 secs. We also tried using a squared term for the derivation of the function, but it led to the code getting stuck between a few values.
4. Tried using multiprocessing to pool all-or-nothing assignments based on different origins, but the code didn't work with the rest of the User Equilibrium assignment.
5. Implemented priority queues for shortest path assignment, maybe the implementation was wrong because it increased the time to 280. So, we ditched it.

AEC after 10000 iterations (Bisection Method): 0.000394
AEC after 10000 iterations (Newton's Method): 0.000280
Time Taken for 10000 iterations (Bisection Method): 197.50 secs

Time Taken for 10000 iterations (Newton's Method): 180.66 secs

The link flows and associated costs after 10000 iterations by using a version of Newton's method are given in the table below:

Franke-Wolfe Method			
Tail Node	**Head Node**	**Link Flows**	**Associated Costs**
1	2	4494.848	6.000816
1	3	8119.27	4.008692
2	1	4519.293	6.000834
2	6	5967.363	6.573626
3	1	8094.825	4.008587
3	4	14007.59	4.269496
3	12	10023.74	4.020191
4	3	14031.8	4.271363
4	5	18007.57	2.315458
4	11	5201.007	7.134178
5	4	18031.87	2.317164
5	6	8797.968	9.997409
5	9	15781.62	9.652298
6	2	5991.807	6.59957
6	5	8806.151	10.01975
6	8	12492.75	14.69025
7	8	12101.54	5.552172
7	18	15794.84	2.062239
8	6	12525.38	14.82334
8	7	12041.45	5.501854
8	9	6882.847	15.17525
8	16	8388.22	10.72813
9	5	15797.74	9.671334
9	8	6836.932	15.03854
9	10	21744.25	5.682617
10	9	21814.45	5.717429
10	11	17726.29	12.40512
10	15	23126.22	13.72294
10	16	11046.96	20.08401
10	17	8100.058	16.30825
11	4	5300.91	7.223865
11	10	17604.11	12.20306
11	12	8365.426	13.59074
11	14	9776.144	13.69138
12	3	9975.088	4.019801
12	11	8404.969	13.73528
12	13	12289.13	3.022808
13	12	12380.02	3.02349
13	24	11121.21	17.66029

14	11	9814.325	13.84367
14	15	9036.403	12.23456
14	23	8400.272	9.078946
15	10	23192.59	13.81197
15	14	9080.166	12.37573
15	19	19083.35	4.32623
15	22	18409.08	9.087018
16	8	8406.668	10.77868
16	10	11072.93	20.23581
16	17	11694.34	9.499752
16	18	15279.33	3.163508
17	10	8100.099	16.30842
17	16	11683.13	9.471042
17	19	9952.883	7.436324
18	7	15854.93	2.063191
18	16	15334.96	3.165902
18	20	18978.13	4.259444
19	15	19116.66	4.335513
19	17	9941.716	7.411968
19	20	8688.11	9.458419
20	18	18993.85	4.260305
20	19	8710.251	9.514271
20	21	6302.218	8.165929
20	22	7000.291	7.713581
21	20	6240.26	8.082003
21	22	8619.484	4.213449
21	24	10309.01	11.92267
22	15	18385.91	9.056424
22	20	7000.112	7.713304
22	21	8607.235	4.200893
22	23	9661.494	12.36466
23	14	8394.69	9.06546
23	22	9625.89	12.24204
23	24	7902.917	3.759245
24	13	11112.11	17.61561
24	21	10259.3	11.75181
24	23	7861.732	3.722858

Table 3.1

After comparing all the results, the Franke-Wolfe method using this version of Newton's method to calculate the step size converges fastest to the solution. We also ran the code for Anaheim to find that Newton's works faster than Bisection.