TRAFFIC ENGINEERING OF TELECOMMUNICATION NETWORKS

Group 6: João Coelho(80335) and Javier Borrallo Fernández(92092)
Desempenho e Dimensionamento de Redes
4º MIECT

João Coelho(80335) Javier Borrallo Fernández(92092)

1st Assignment - Minimizing Worst Link Load

Just by changing improving a bit of the code that was given to us (checking every neighbour in each flow), we got better results than before. These are the results that we got:

(i) The obtained worst link load for k= 5, 10, 15, 20, 25 and 30.
The obtained worst Link load is the same for every K value.
0.6410
(ii) the total local search running time.

K	Elapsed Time(s)
5	4.975863
10	10.556326
15	17.728788
20	19.346038
25	25.342628
30	30.283618

What is the value of k that gives the best performance?
The best performance is given by K=5, as it reaches the same Worst Link Load as the other K values, but in less time, 4.98s.

2nd Assignment
Minimizing Energy Consumption for a Given Worst Link Load

For this assignment, we developed an algorithm based on 2 selection criteria:
(i) the link providing the lowest worst link load and

W	Lowest worst link load			
	x	y	consumption	Worst link load
0.85	4.0000	5.0000	38.0000	0.5970
	1.0000	11.0000	36.0000	0.5970
	3.0000	4.0000	35.0000	0.6390
	10.0000	12.0000	34.0000	0.6520
	13.0000	15.0000	33.0000	0.6690
	7.0000	8.0000	29.0000	0.7560
	8.0000	11.0000	28.0000	0.7860
	1.0000	2.0000	27.0000	0.7790
0.8	4.0000	5.0000	38.0000	0.5970
	1.0000	11.0000	36.0000	0.5970
	3.0000	4.0000	35.0000	0.6390
	10.0000	12.0000	34.0000	0.6520
	13.0000	15.0000	33.0000	0.6690
	7.0000	8.0000	29.0000	0.7560
	8.0000	11.0000	28.0000	0.7860
	1.0000	2.0000	27.0000	0.7790
0.75	4.0000	5.0000	38.0000	0.5970
	1.0000	11.0000	36.0000	0.5970
	3.0000	4.0000	35.0000	0.6390
	10.0000	12.0000	34.0000	0.6520

	13.0000	15.0000	33.0000	0.6690
0.7	4.0000	5.0000	38.0000	0.5970
	1.0000	11.0000	36.0000	0.5970
	3.0000	4.0000	35.0000	0.6390
	10.0000	12.0000	34.0000	0.6520
	13.0000	15.0000	33.0000	0.6690

(ii) the link providing the best ratio of the worst link load divided by the energy consumption of the link.

W	Link providing the best ratio			
	x	y	consumption	Worst link load
0.85	8.0000	9.0000	35.0000	0.7210
	5.0000	7.0000	31.0000	0.7350
	1.0000	11.0000	29.0000	0.7350
	1.0000	2.0000	28.0000	0.6450
	6.0000	12.0000	26.0000	0.7670
	10.0000	15.0000	25.0000	0.7210
	8.0000	11.0000	24.0000	0.7530
	3.0000	4.0000	23.0000	0.7950
0.8	8.0000	9.0000	35.0000	0.5970
	1.0000	11.0000	36.0000	0.7210
	5.0000	7.0000	31.0000	0.7350
	1.0000	11.0000	29.0000	0.7350
	1.0000	20000	28.0000	0.6450
	6.0000	12.0000	26.0000	0.7670
	10.0000	15.0000	25.0000	0.7210
	8.0000	11.0000	24.0000	0.7530

	3.0000	4.0000	23.0000	0.7950
0.75	8.0000	9.0000	35.0000	0.7210
	5.0000	7.0000	31.0000	0.7350
	1.0000	11.0000	29.0000	0.7350
	1.0000	2.0000	28.0000	0.6450
	10.0000	12.0000	27.0000	0.6330
	10.0000	15.0000	26.0000	0.6670
	8.0000	11.0000	25.0000	0.7240
	3.0000	4.0000	24.0000	0.7450
0.7	1.0000	11.0000	37.0000	0.6410
	5.0000	11.0000	35.0000	0.6830
	8.0000	9.0000	31.0000	0.6890
	1.0000	5.0000	30.0000	0.6810
	15.0000	17.0000	29.0000	0.6870

We can observe that, using our algorithm, we have the best results considering the link providing the best ratio of the worst link load divided by the energy consumption of the link although it’s not so much different from the first algorithm.

In the first case, the worst link load is related to the consumption as the lower the worst link load is, the higher is the consumption of that path.

Using the second algorithm, we find that even though the results on the worst link loads are not so low as expected, we see some really good improvements on some links that provide low consumption and low link load (i.e. x=1 and y=2).

ANNEX I – Computing solutions for the link providing the lowest worst link load

```
while(worstlinkload < w)
    worstlinkload= inf;
    for i= 1:nNodes-1
        for j= 1+i:nNodes
            if C(i,j) > 0
                auxR= R(i,j);
                auxL= L(i,j);
                R(i,j)= 0;
                R(j,i)= 0;
                L(j,i)= inf;
                L(i,j)= inf;
                consump= sum(sum(C))/2-C(i,j);
                auxLocalSearch= localSearch(R,L,T);
                if auxLocalSearch < worstlinkload
                    worstlinkload= auxLocalSearch;
                    bestI= i;
                    bestJ= j;
                    solution= [i j consump worstlinkload];
                end
                R(i,j)= auxR;
                L(i,j)= auxL;
                R(j,i)= auxR;
                L(j,i)= auxL;
            end
        end
    end
    finalSolution= [finalSolution; solution]
    R(bestI, bestJ)= 0;
    R(bestJ, bestI)= 0;
    C(bestI, bestJ)= 0;
    C(bestJ, bestI)= 0;
    L(bestJ, bestI)= inf;
    L(bestI, bestJ)= inf;
end
```

ANNEX II – Computing solutions for the link providing the best ratio of the worst link load divided by the energy consumption of the link

```
while(auxLocalSearch < w)
    ratio= 1.1;
    for i= 1:nNodes-1
        for j= 1+i:nNodes
            if C(i,j) > 0
                auxR= R(i,j);
                auxL= L(i,j);
                R(i,j)= 0;
                R(j,i)= 0;
                L(j,i)= inf;
                L(i,j)= inf;
                consump= sum(sum(C))/2-C(i,j);
                auxLocalSearch= localSearch(R,L,T);
                auxRatio= auxLocalSearch/C(i,j);
                if auxRatio < ratio && auxLocalSearch <= w
                    bestI= i;
                    bestJ= j;
                    ratio= auxRatio;
                    solutionRatio= [i j consump auxLocalSearch]
                end
                R(i,j)= auxR;
                L(i,j)= auxL;
                R(j,i)= auxR;
                L(j,i)= auxL;
            end
        end
    end
    secondFinalSolution= [secondFinalSolution; solutionRatio]
    R(bestI, bestJ)= 0;
    R(bestJ, bestI)= 0;
    C(bestI, bestJ)= 0;
    C(bestJ, bestI)= 0;
    L(bestJ, bestI)= inf;
    L(bestI, bestJ)= inf;
end
```