FRTN30 Network Dynamics: Hand-In 2

Alfred Bornefalk, al1718bo-s@student.lu.se, 2022-05-05

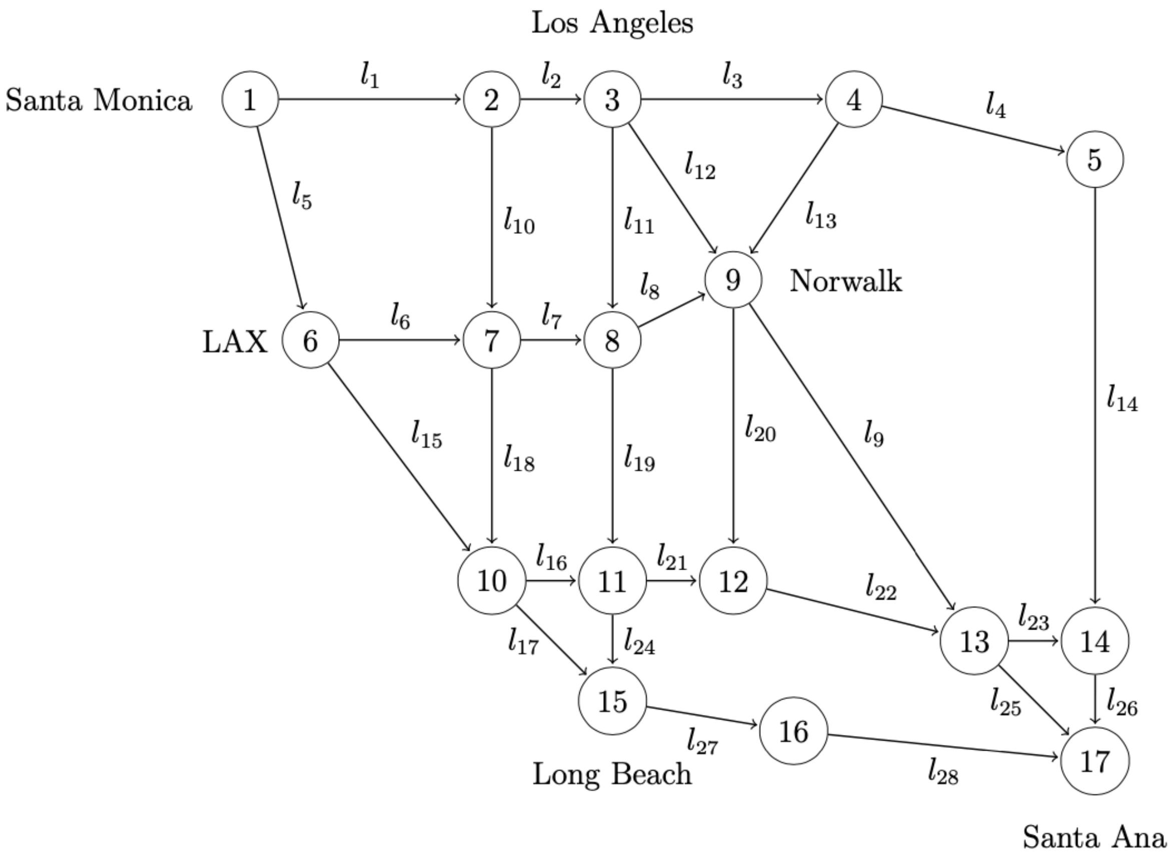

Table of Contents

1. Traffic Tolls in Los Angeles

In this assignment, we are to study traffic flows on the highway network in Los Angeles. For simplifying purposes, an approximate highway map will be examined, which covers parts of the real highway network. The cut covers 17 nodes and their possible paths all the way from Santa Monica (node 1) to Santa Ana (node 17). In total, there are 28 links.

The node-link incidence matrix *B* for the traffic network in question is given, where the rows are associated with the nodes of the network, whereas the columns represent the links. The maximum flow capacity of each link e_i is given by the corresponding element in vector C_e. Additionally, the minimum traveling times are given as a vector l_e.

Before moving to the first subsection, we note that the following delay function is defined for each link:

$$d_e(f_e) = \frac{l_e}{1-\frac{f_e}{C_e}}, 0 \le f_e < C_e$$

For $f_e \ge C_e$, the value of the delay function is considered to be positive infinity.

1.1. The Shortest Path

Here, we are asked to find the shortest path between node 1 and 17. (Santa Monica and Santa Ana, that is.) Equivalently, we are going to find the fastest path in an empty network.

To carry out this analysis, we first define the source node and the target node for each link, i.e. we associate l_1 with source node 1 and target node 2, l_2 with 2 and 3, et cetera. Combining this with the given travel times gives the graph in figure 1 below.

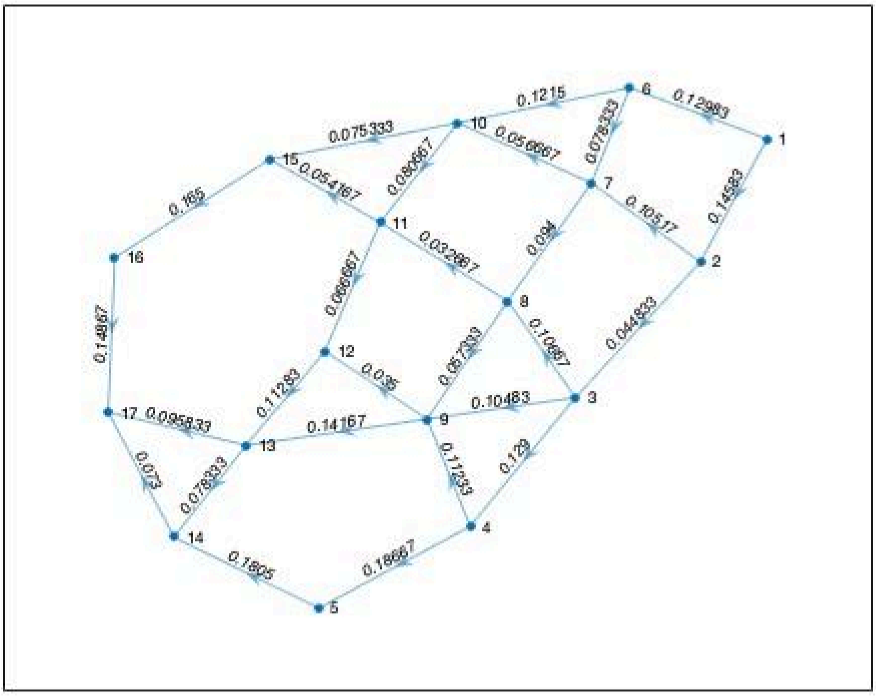

Figure 1: Simplified representation of the Los Angeles traffic network. The value on the nodes represent the corresponding place, whereas the value on the links represent the associated travel time.

From figure 1, we can see (as one expects) that the graph is weighted. Defining *G1* as the network seen in figure 1, and using node 1 as the source node, and 17 as the target node, running the MATLAB command

```
shortestpath(G1, 1, 17)
```

yields that the shortest path from 1 to 17 is through the nodes 2, 3, 9, and 13 (in that order).

1.2. The Maximum Flow

Now, our aim is to find the maximum flow between node 1 and 17. More precisely, we aim to determine the maximum throughput *v* from Santa Monica to Santa Ana without violating the given link capacity constraints. Thus, the optimization problem under consideration is

$$v^*_{1,17} = \max v \text{ subject to}$$
$$v \geq 0, 0 \leq f \leq c, Bf = v(\delta^{(1)} - \delta^{(17)})$$

To investigate this problem, we combine the previously defined nodes and links with the given (maximum) capacities, resulting in the network depicted in figure 2 below.

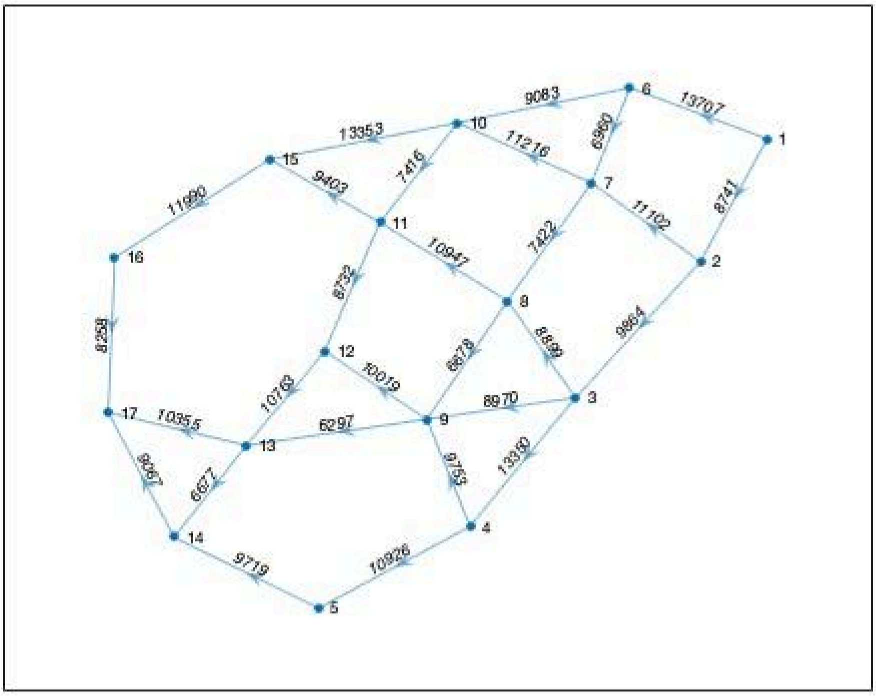

Figure 2: Simplified representation of the Los Angeles traffic network. The value on the nodes represent the corresponding place, whereas the value on the links represent the associated capacity.

Observing figure 2, it is evident that the different links do bear different capacities. Defining *G2* as the network seen in figure 2, again using node 1 as the source node, and 17 as the target node, running the MATLAB command

```
maxflow(G1, 1, 17)
```

yields that the maximum flow from 1 to 17 is 22,448. An interesting observation is that this is also the maximum flow from 1; 8,741 can be sent to node 2, and 13,707 to node 6. Alas, the bottleneck for the maximum flow is the link capacities from our source (Santa Monica).

1.3. The Exogenous Inflow or Outflow at Each Node

In this subproblem, we are requested to, given a flow vector for our simplified Los Angeles traffic network, compute the external inflow or outflow at each node. On our hands, we have an exogenous net flow, i.e. given a vector *v*, the constraint

$$\sum_{i=1}^{17} v_i = 0$$

is satisfied. Given a net flow on node *i*, the exogenous inflow is defined as

$$[v_i]_+ = max\{0, v_i\}$$

whilst the exogenous outflow is given by

$$[v_i]_- = max\{0, -v_i\}.$$

For illustrative purposes, we combine the previously defined nodes and links with the given flow, thereby creating the network shown in figure 3 below.

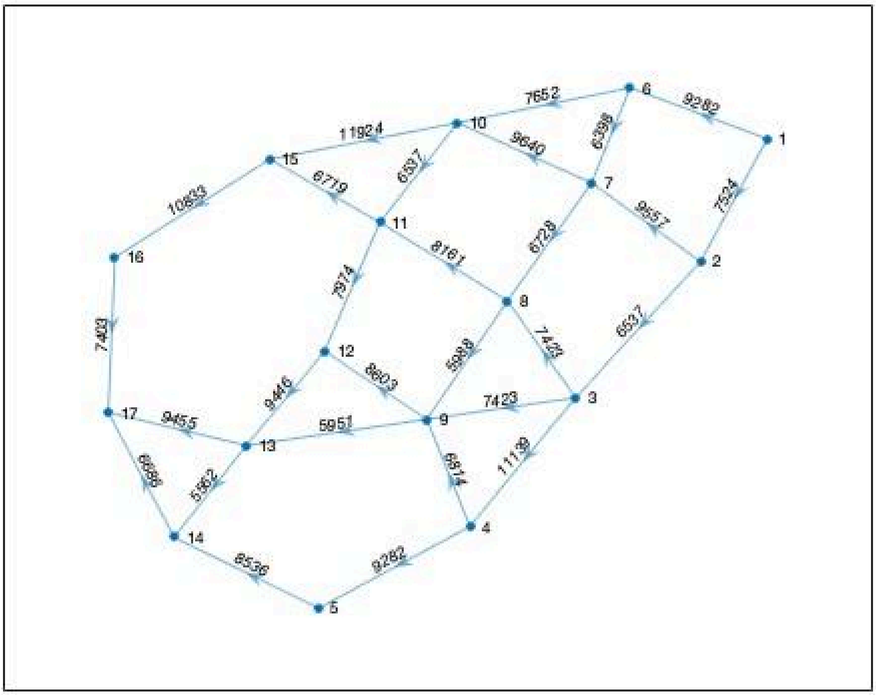

Figure 3: Simplified representation of the Los Angeles traffic network. The value on the nodes represent the corresponding place, whereas the value on the links represent the associated flows.

Now, to determine the exogenous inflow or outflow for each node, we proceed with multiplying the given traffic vector with the aforementioned capacity vector. The results are presented in table 1.

Table 1: The exogenous inflow or outflow at each node in the simplified Los Angeles traffic network.

Node	Inflow	Outflow
1	16,806	0
2	8,570	0
3	19,448	0
4	4,957	0
5	0	746
6	4,768	0

7	413	0
8	0	2
9	0	5,671
10	1,169	0
11	0	5
12	0	7,131
13	0	380
14	0	7,412
15	0	7,810
16	0	3,430
17	0	23,544

Observing table 1, we can see that seven nodes have an exogenous inflow, whilst ten has an outflow. Furthermore, Santa Monica (node 1) has the second largest inflow, and Santa Ana (node 17) has the most prevalent outflow.

1.4. The Social Optimum

In this subproblem, we are asked to find the social optimum f^* with respect to the delays on the different links $d_e(f_e)$. More specifically, we are to minimize the cost function

$$\sum_{e\in\varepsilon} f_e d_e(f_e) = \sum_{e\in\varepsilon} \frac{f_e l_e}{1-\frac{f_e}{C_e}} = \sum_{e\in\varepsilon} \left(\frac{l_e C_e}{1-\frac{f_e}{C_e}} - l_e C_e\right)$$

subject to the flow constraints. In order to solve this optimization problem, we will use the MATLAB-based modeling system for convex optimization: CVX. The travel time l and the capacities c are known. Thus, we find the social optimum traffic flow, with f^* denoted as f^* and M being the total number of links in the network (28), through the algorithm

```
cvx_begin
    variable f(M)
    minimize sum(traveltime .* capacities .* inv_pos(1 - f ./ capacities) - traveltime .*
    capacities)
    subject to
        traffic * f == lambda - mu;
        0 <= f <= capacities;
cvx_end
```

This numerical optimization resulted in the social optimum flows presented in figure 4.

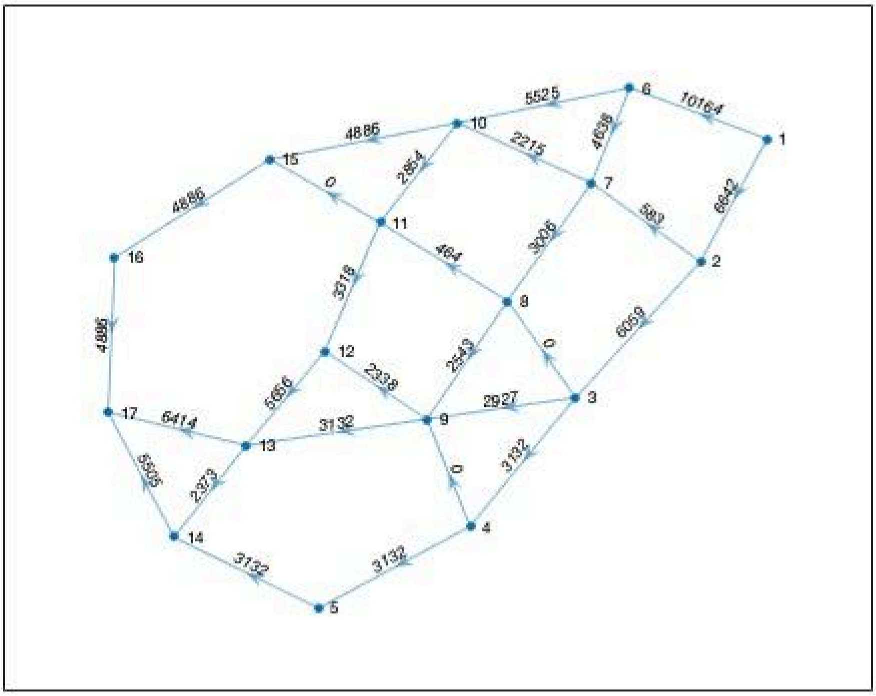

Figure 4: Simplified representation of the Los Angeles traffic network. The value on the nodes represent the corresponding place, whereas the value on the links represent the social optimum flows.

Alas, figure 4 displays the social optimum flows, i.e. the Los Angeles traffic assignment that minimizes the total travel costs. Comparing figure 4 to figure 3, we can see that in the former, the outflow from node 1 is 6,642 and 10,164, whereas it is 7,524 and 9,282 in the latter; both of these flows sum up to 16,806. Some links are even assigned a flow of zero in the social optimum, namely from node 3 to 8, 11 to 15, and 4 to 9.

1.5. The Wardrop Equilibrium

At this point, we aim to find the Wardrop equilibrium $f^{(0)}$. In our case, $f^{(0)}$ is associated with the *1-17* path distribution *z* such that, if some drivers choose path *p* as their route from Santa Monica to Santa Ana, then the total delay associated with this path cannot be worse than the total delay associated with any other *1-17* path q. The interpretation of this is the result of a rational, and selfish, driver behavior, where no one would choose a sub-optimal route if a better one is available. To find said equilibrium, we use the following cost function:

$$\sum_{e \in \varepsilon} \int_{0}^{f_e} d_e(s)ds.$$

Again, since all other inputs are known, the parameter that we want to determine in order to minimize the cost function above is the flow vector *f*. Denoting it with *g* in MATLAB, CVX solves this optimization problem with the algorithm

```
cvx_begin
	variable g(M)
	minimize sum(-traveltime .* capacities .* log(1 - g ./ capacities))
	subject to
		traffic * g == lambda - mu;
		0 <= g <= capacities;
cvx_end
```

This numerical optimization resulted in the Wardrop equilibrium flows presented in figure 5.

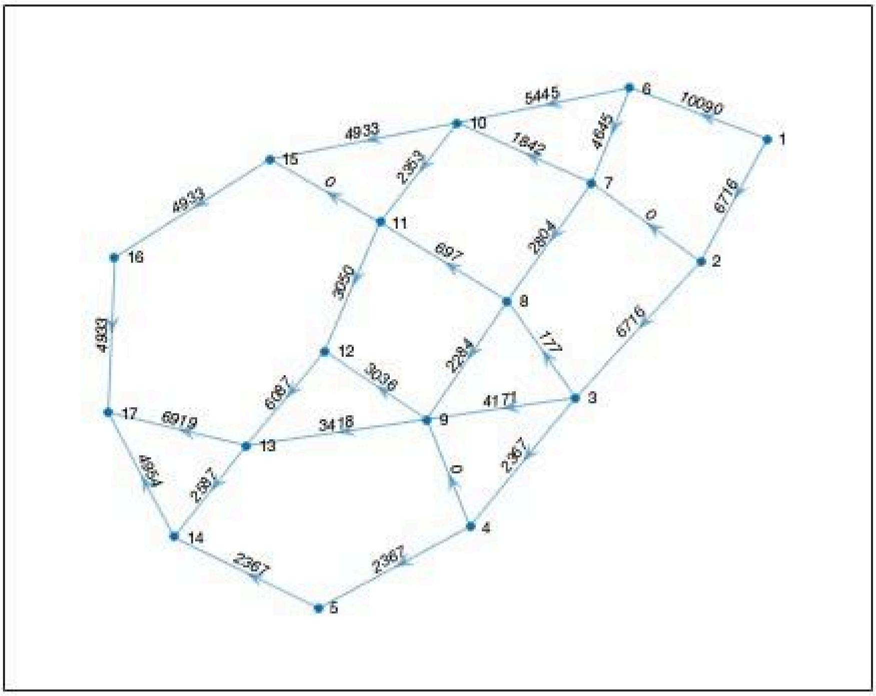

Figure 5: Simplified representation of the Los Angeles traffic network. The value on the nodes represent the corresponding place, whereas the value on the links represent the Wardrop equilibrium flows.

Expectedly, figure 5 displays similar behavior to figure 4. For instance, the flow is still zero from node 4 to 9 and 11 to 15. Additionally, the flow from node 1 to 6 has decreased with 74. (Equivalently, since Santa Monica is our source node and we wish to utilize its full capacity, the flow from node 1 to 2 has increased with 74.) However, some minor discrepancies arise. Instead of no flow (i.e. zero) from node 3 to 8, we now have a flow of 177. Furthermore, the flow of 583 from node 2 to 7 in figure 4 has been reduced to zero in the Wardrop equilibrium.

1.6. The New Wardrop Equilibrium

Here, we introduce a modification to the system; tolls such that the toll on link *e* is

$$\omega_e = f_e^* d'_e(f_e^*),$$

are created. where f_e* is the flow at the system optimum. Hence, the delay on link e is now given by

$$d_e(f_e) + \omega_e,$$

which is why we are interested in computing the new Wardrop equilibrium $f^{(\omega)}$.

Once again, the parameter that we want to determine in order to minimize the cost function above is the flow vector f*. Denoting it with h in MATLAB, and also including the above mentioned tolls, CVX solves this optimization problem with the algorithm

```
cvx_begin
    variable h(M)
    minimize sum(-traveltime .* capacities .* log(1 - h ./ capacities) + omega1 .* h)
    subject to
        traffic * h == lambda - mu;
        0 <= h <= capacities;
cvx_end
```

This numerical optimization resulted in the new Wardrop equilibrium flows below.

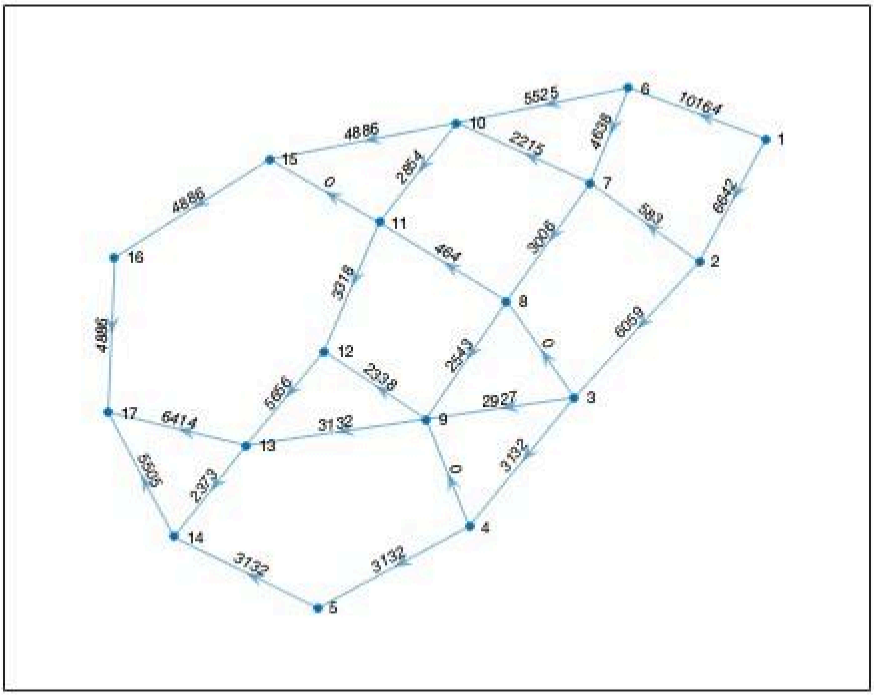

Figure 6: Simplified representation of the Los Angeles traffic network. The value on the nodes represent the corresponding place, whereas the value on the links represent the new Wardrop equilibrium flows.

Whilst observing figure 6, an interesting conclusion can be drawn. Namely, that the optimum flows under the new Wardrop equilibrium are exactly equal to (apart from a few decimal points not seen in the plots) the system optimum *before* the tolls were introduced. Thus, with the introduction of the tolls, said rational, and selfish, drivers no longer gain anything by deviating from the system optimum.

1.7. The New System Optimum

In the last part of this assignment, instead of the total delay, we let the cost be the total additional delay compared to the total delay in free flow be given by

$$c_e(f_e) = f_e(d_e(f_e) - l_e).$$

(Naturally, still subject to the flow constraints.) Now, we wish to compute the new system optimum f^* for said costs. Apart from using a built-in function in CVX, we will construct tolls ω_e^* in such a manner that the new Wardrop equilibrium with the constructed tolls $f^{(\omega^*)}$ coincides with f^*. In order to verify our solution, the new Wardrop equilibrium in question will also be computed.

We start by computing the new system optimum f^*. Denoting it with p in MATLAB, and letting p_1 describe the total cost of travel, CVX solves this optimization problem with the algorithm

```
cvx_begin
	variable p(M)
	p1 = 0;

	for k = 1:M
		p1 = p1 + traveltime(k) + quad_over_lin(p(k), capacities(k) - p(k));
	end

	minimize p1
	subject to
		traffic * p == lambda - mu;
		0 <= p <= capacities;
cvx_end
```

The result of this numerical optimization is presented in figure 7.

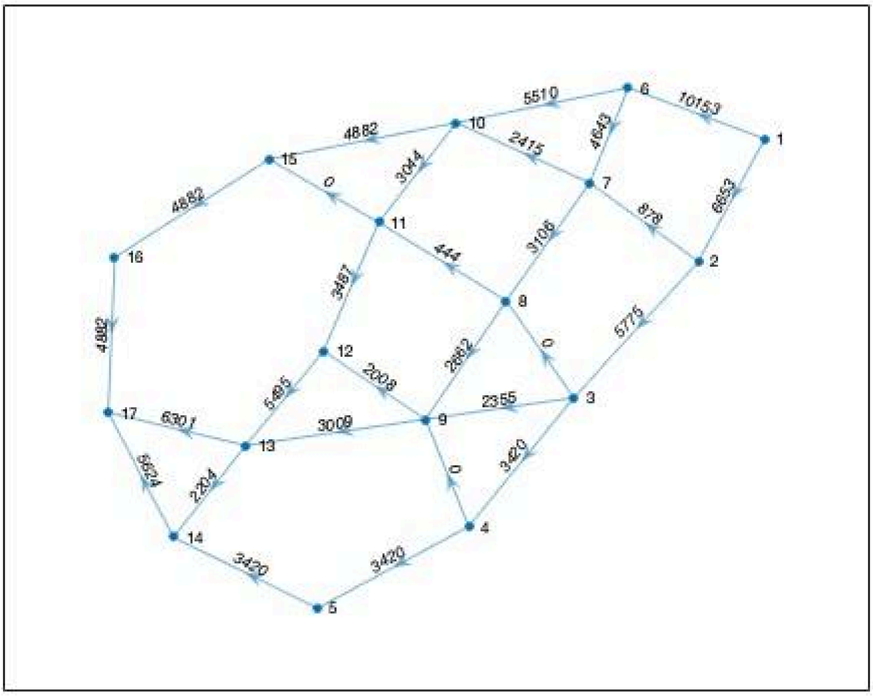

Figure 7: Simplified representation of the Los Angeles traffic network. The value on the nodes represent the corresponding place, whereas the value on the links represent the new social optimum flows.

Lastly, we verify the results in figure 7 with the following CVX-algorithm (letting q denote the new Wardrop equilibrium $f^{(\omega^*)}$):

```
cvx_begin
        variable q(M)
        minimize sum(-traveltime .* capacities .* log(1 - q ./ capacities) + omega2 .* q)
        subject to
                traffic * q == lambda - mu;
                0 <= q <= capacities;
cvx_end
```

Here, we designed the tolls in a similar manner to the previous subsection; the difference being that the travel time on each link e was subtracted. The graphical output of the algorithm is finally presented in figure 8.

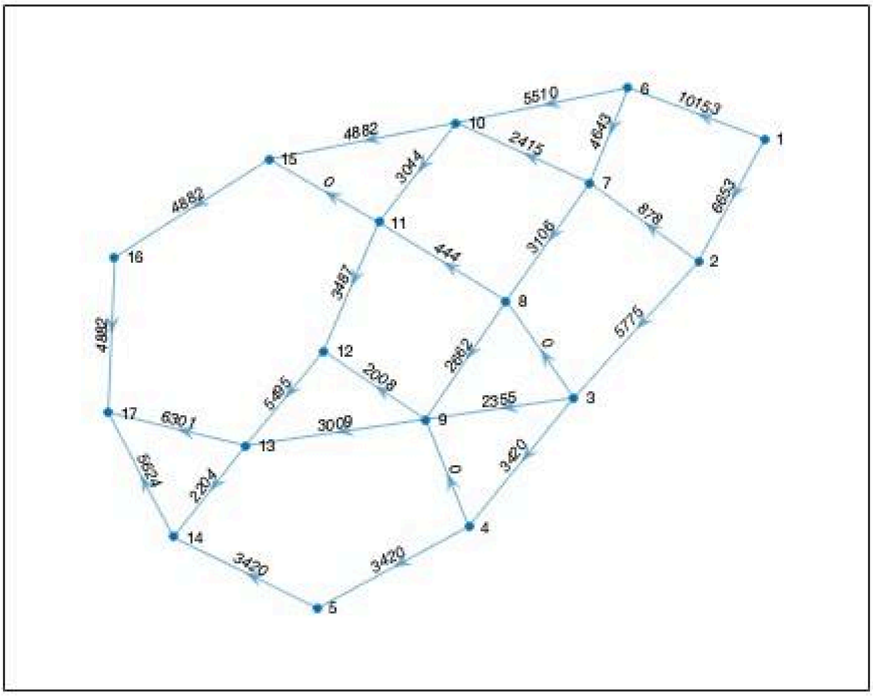

Figure 8: Simplified representation of the Los Angeles traffic network. The value on the nodes represent the corresponding place, whereas the value on the links represent the new Wardrop optimum flows (the tolls designed in a way to equal the values in figure 7).

Comparing figure 7 and figure 8, the verification process is concluded. The (optimum) flows in the Los Angeles traffic network now equal each other, and there is still no travel (a flow of zero) from the nodes 4 to 9, 3 to 8, and 11 to 15, which was the case in figure 6, where we first introduced tolls. One last observation that is worthy of mentioning is the fact that the flow from node 2 to 7 first increased from zero (the Wardrop equilibrium with no tolls), then to 583 (the first toll design), and finally to 878 (the case of figure 7 and 8). This clearly indicates that introducing tolls on different links (roads, that is) may yield an effect on the flow of traffic. Queuing times in the Los Angeles traffic are known to be extraordinary; the responsible politicians ought to learn a thing or two from studying network dynamics when deciding on pricing mechanisms for the roads.