Optimization of Formula 1 Racing

Nassim Talbi

Advisor: Dr. Darin Mohr

January 9, 2015

Abstract

In Formula One Racing, cars make hairpin turns at high speeds. Those turns must be executed perfectly in order for the car to minimize its total race time. We model a 180 degree turn and consider the turn as a network flow on a grid, where the nodes represent the positions along the turn and each edge in the network flow represents the path it takes to reach the next node. We use the computer program AMPL to find the optimal solution path within the network.

1 Introduction

A network flow is a directed graph where each edge is assigned a specific weight [1]. Quantities move between nodes along edges starting from source nodes and ending at sink nodes. In the model we are creating, the source node represents the starting position of the car. Sink nodes are the nodes that mark the end of the network. Therefore the sink nodes are located at the end of the turn and represent the possible end positions of the car. Transshipment nodes are possible nodes that the car can move between when traveling from the source to the sink nodes. In our model, the transshipment nodes are encompassed within the race track between the source and the sink nodes. The car moves between these nodes on weighted edges, where each edge represents the cost of moving to each individual node. Known as the minimum cost network flow problem, this takes into account each of the weighted edges. Our goal is to find a route for the car to travel that minimizes time.

Solving a general minimum cost network flow problem will allow us to find the route the car must take that minimizes distance. In general, the movement of the car can be modeled by:

$$\text{Minimize} \quad \sum_{i,j \in H} C_{ij} X_{ij}.$$

Subject to:

$$\sum_{i \in N} X_{ij} - \sum_{i \in N} X_{ji} = -1, \text{ for each j} \in \text{So}$$

$$\sum_{i \in N} X_{ij} - \sum_{i \in N} X_{ji} = 1, \text{ for each j} \in \text{Si}$$

$$\sum_{i \in N} X_{ij} - \sum_{i \in N} X_{ji} = 0, \text{ for each j} \in \text{T}$$

where So is the set of source nodes, Si is the set of sink nodes, T is the set of transshipment nodes, H is the set of all edges and N is the set of all nodes in the directed graph. The objective function,

$$\text{Minimize} \quad \sum_{i,j \in H} C_{ij} X_{ij}$$

is the sum across all weighted edges $(i, j) \in H$. C_{ij} represents the cost of travel from node i to j. X_{ij} represents whether that edge is used or not [2]. The first set of constraints,

$$\sum_{i \in N} X_{ij} - \sum_{i \in N} X_{ji} = -1, \text{ for each j} \in \text{So}$$

is the source constraints. For source nodes, the variable X_{ij} has the property that the flow out of the node exceeds the flow into the node. For our program, we are considering one car so that the flow out is one more than the flow in. The second set of constraints,

$$\sum_{i \in N} X_{ij} - \sum_{i \in N} X_{ji} = 1, \text{ for each j} \in \text{Si}$$

are the sink constraints. For sink nodes, the variable X_{ij} is the opposite of the source node: it has the property that the flow into the node exceeds the flow out. For our program, again we are considering one car so that the flow in is one more than the flow out. The third set of constraints,

$$\sum_{i \in N} X_{ij} - \sum_{i \in N} X_{ji} = 0, \text{ for each j} \in \text{T}$$

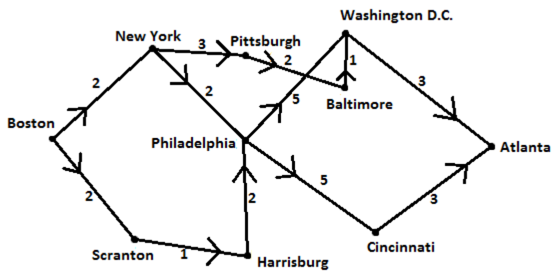
are the transshipment constraints. Note that for transshipment nodes, the variable satisfies conservation of flow, meaning the flow into node i equals the flow out to node j.

2 Model

To model a turn on a race track, we place a grid system on top of a track. Each node is a position on the track. The edges represent the time in seconds needed to move between each node. The optimal solution is the minimal route around the turn that is completed in the shortest amount of time. Using the computer program AMPL, we can optimize any objective function subject to constraints to find an optimal solution.

2.1 First Model in AMPL

Consider the following: a truck needs to move supplies from Boston to Atlanta. The truck wants to take the shortest path possible. Each edge weight represents the amount of time it takes for the truck to travel between cities. The model has 10 nodes with a source at Boston and a sink at Atlanta, with 12 edges. We seek the optimal path from Boston to Atlanta

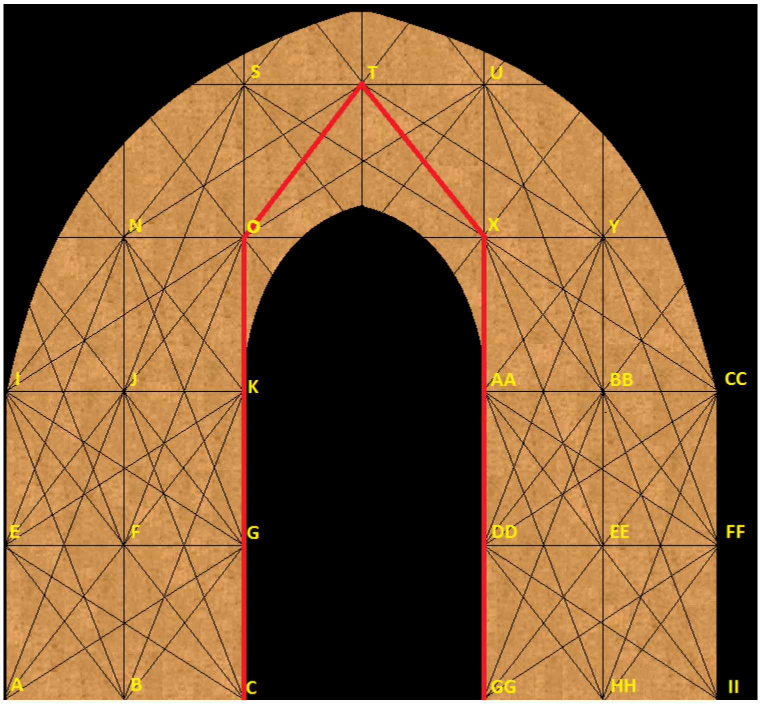


which minimizes time.

Using the constraints for source, sink, and transshipment nodes and keeping the objective function the same as mentioned in the introduction, we use AMPL to find the optimal solution. The weight of each edge represents the hours it takes to drive from city to city. The optimal path is Boston - New York - Pittsburgh - Baltimore - Washington D.C. - Atlanta for a total of 11 hours.

2.2 25 Point Grid in AMPL

Next we construct a model that is similar to the problem proposed. Note that the directed graph is missing the direction arrows because each route can be traversed in either direction. The constraint forces the car to begin at the source node and finish at the sink node. The model has 25 nodes.



We assume a unit grid, therefore edge connections A to E, B to F, F to G, etc. were assigned a weight of 1 unit. Furthermore, edge connections A to F, B to G, B to E, etc. were assigned a weight of $\sqrt{2}$. We assume one unit represents ten meters.

2.3 Refined 25 Point Grid in AMPL

Now we consider a similar model with the same amount of nodes and more edge connections. We can include more edges when we consider the edge path two spaces away in the taxicab norm. We also incorporate velocity into the objective function:

$$\text{Minimize} \quad \sum_{i,j \in H} \frac{C_{ij} X_{kj}}{S_{kj}}.$$

S_{ij} represents the speed of the movement across an edge.For this model, we allow the car to travel at variable speed. Incorporating velocity, now the program minimizes time for the car to complete the turn. We include a constraint on velocity:

$$\sum_{i,j \in H} C_{ij} X_{kj} S_{kj} \leq \text{vmax}$$

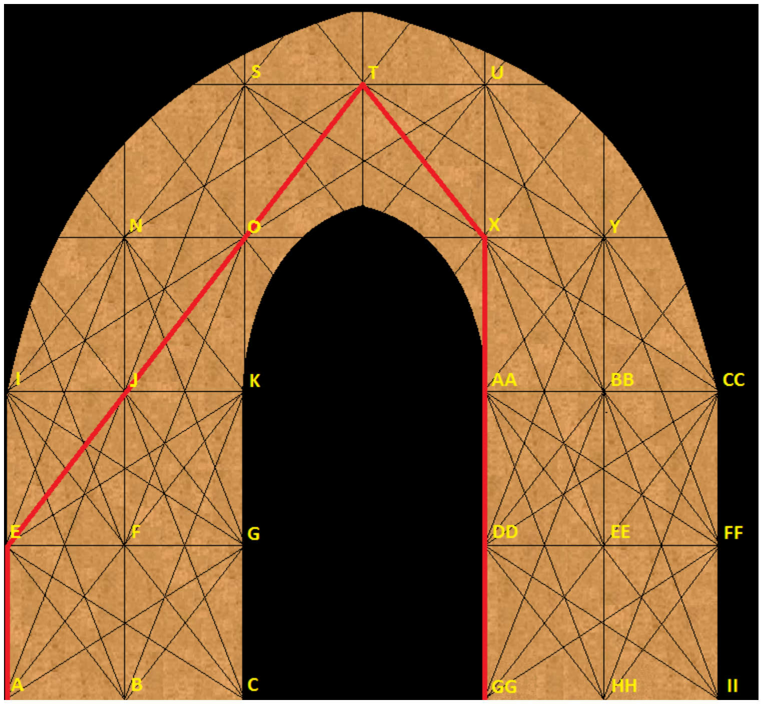
The constraint is called vmax. The purpose of this addition to the constraint is to make the program realistic. The car now has an upper bound on speed and this constraint enforces that. We find the max velocity of a Formula 1 car to be 223 mph [3]. We convert that to meters per second (since the turn is in meters) and calculate a max velocity (vmax) of 99.6889 meters per second. This allows the car to travel at any speed 0-99.6889 meters per second.

3 Assumptions

We model the 180 degree turn on a grid system. We give the program two starting options (A and C) to see which one gives us the optimal solution. We assume that the car can end at any three of the three sink nodes (GG, HH, II). We assume the max speed of an F1 race car to be 223 mph (99.6889 m/s). Lastly, we also assume that the only routes possible are the routes that follow the track from source node to sink node.

4 Results

The program gives us a solution for claiming the only source node at A, B, and C. The first option starts the car on node A.



The results tell us two things: the speed at which the race car moves from node to node and the optimal path. We convert the velocity back from meters per second into miles per hour.

Table 1: Source Node at A

Edge Connection	Speed (mph)
A-E	223
E-J	157.7
J-O	157.7
O-T	157.7
T-X	157.7
X-AA	223
AA-DD	223
DD-GG	223

The objective is 0.12035. Since each unit is equivalent to ten meters, the car completed the turn in 12.035 seconds. The second option starts the car on node B.

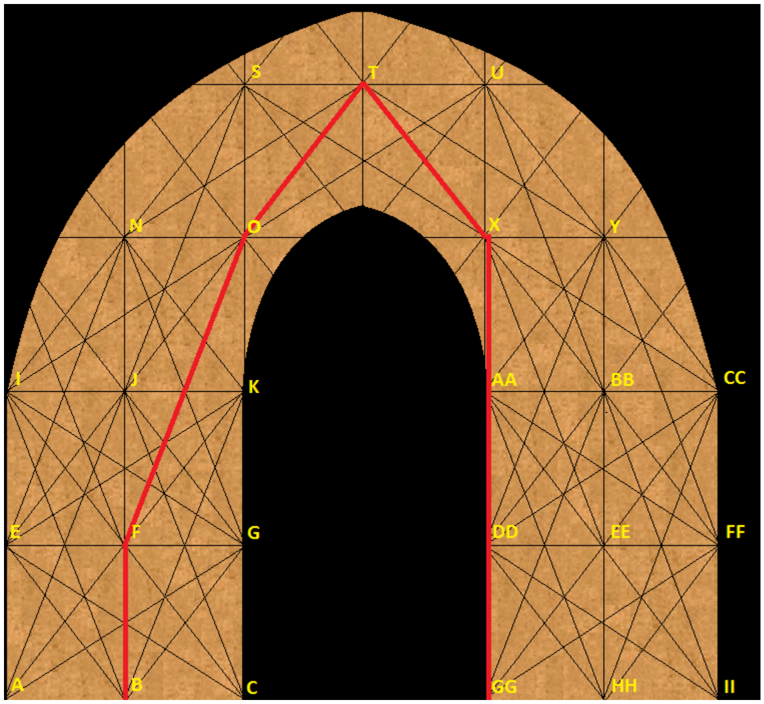


Table 2: Source Node at B

Edge Connection	Speed (mph)
B-F	223
F-O	99.73
O-T	157.7
T-X	157.7
X-AA	223
AA-DD	223
DD-GG	223

The objective is 0.13039. Since each unit is equivalent to ten meters, the car completed the turn in 13.039 seconds. The third option starts the car on node C.

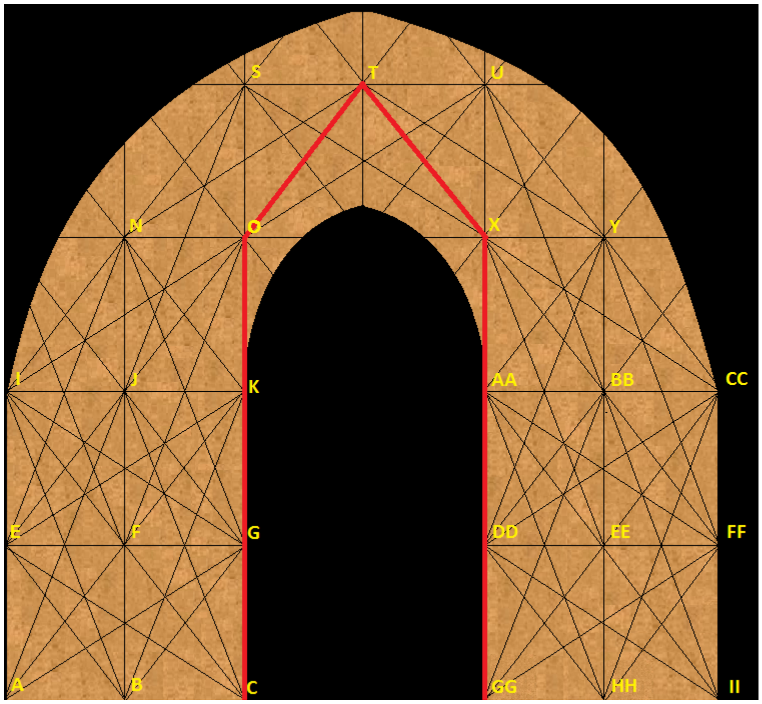


Table 3: Source Node at C

Edge Connection	Speed (mph)
C-O	223
O-T	157.7
T-X	157.7
X-AA	223
AA-DD	223
DD-GG	223

The objective is 0.100306; meaning the car completes the turn in 10.036 seconds. With the source node at C, is found to be the optimal path.

5 Conclusion

After looking at the results, we notice that the smallest objective is when the source node is placed on node C. Though the velocity constraint we placed on the race car only gives the race car an upper bound on speed, it also slows the car down while turning (traveling diagonally). Notice that the speed from node O to T is 149 mph when on other diagonal turns the speed of the car is 157.7. That is the solver we are using believes that after running a numerous amount of iterations (in this case over 19200), it has either reached or come close to the optimal solution. We know that since the model has instantaneous acceleration so the race car travels faster than it would in reality.

In future work, we can continue to expand upon this program. If we add more node points on the grid, it will allow for more edge connections, resulting in a finer graph. Also, we can begin to incorporate physics. We can add centripetal acceleration constraints to model the effect of G-Force on the car during the turn. We already found that the max G-Force to be 4.5 G's on a turn. Finally, with a model of a turn completed, we can move forward and model an entire track.

References

[1] Hillier, Fredrick, and Gerald Lieberman. *Introduction to Operations Research*. 5th ed. New York City: McGraw-Hill, 1990. Print.

[2] Fourer, Robert, and David M. Gay. *AMPL : A Modeling Language for Mathematical Programming*. 2nd ed. Pacific Grove, CA: Thomson/Brooks/Cole, 2003. Print.

[3] Newswire. "F1: V6 Engine cars are much faster on straights." *Speed Network*. Fox Sports, 25 February 2014. Web. *http://www.foxsports.com/speed/formula-1/f1-v6-engine-cars-are-much-faster-on-straights/*, accessed October 3, 2014.