Executive Summary:
Bay Area Destination and Sight-Seeing Tool (B.A.D.A.S.S. Tool Version 2.0)

Background

The genesis of this problem comes from a strong desire to maximize the enjoyment of a road-trip while minimizing time spent driving or stuck in traffic. To satisfy this desire, we've developed a route development tool that will aid in planning a route and recommending destination sights, subject to variable constraints such as minimum destination times, length of trip, and starting time each day. The model can be attacked by various delays such as traffic, congestion, accidents, road construction, and road closures.

Creating the Basic Network

Quickly realizing that the options for a road trip from Monterey are very numerous and that the graph grows much too large for the scale of this project, we selected a short list of destination spots in the San Francisco bay area. The generation of an initial undirected cyclic network is illustrated in Figure 1. The arcs in the overlay are numbered only to aid in creating a more aesthetically appealing network graph. The basic network contains 41 nodes corresponding to destinations and road intersections. Note that 40 and 41 represent the Start and End Nodes at Monterey and are both connected to Nodes 38 and 39.

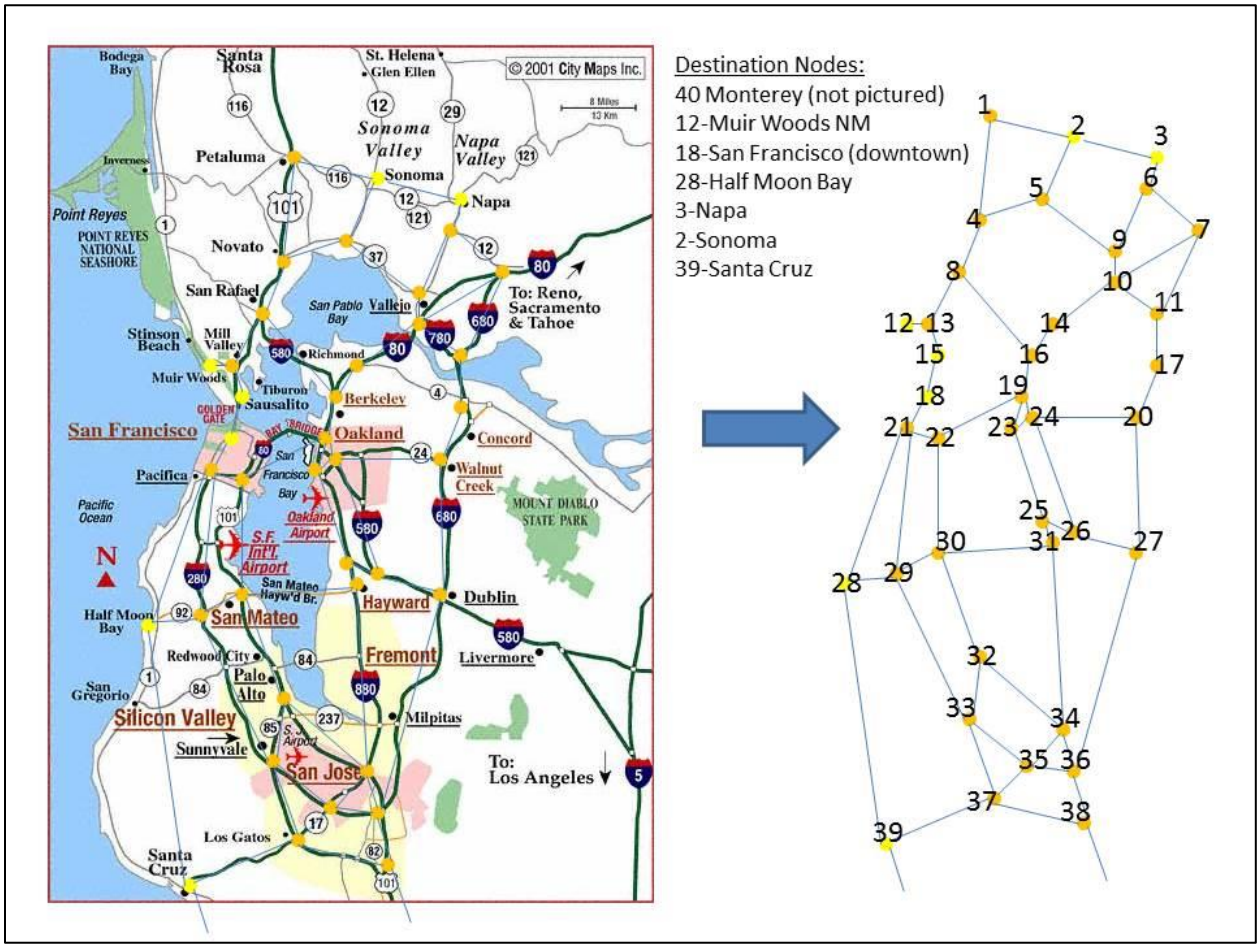
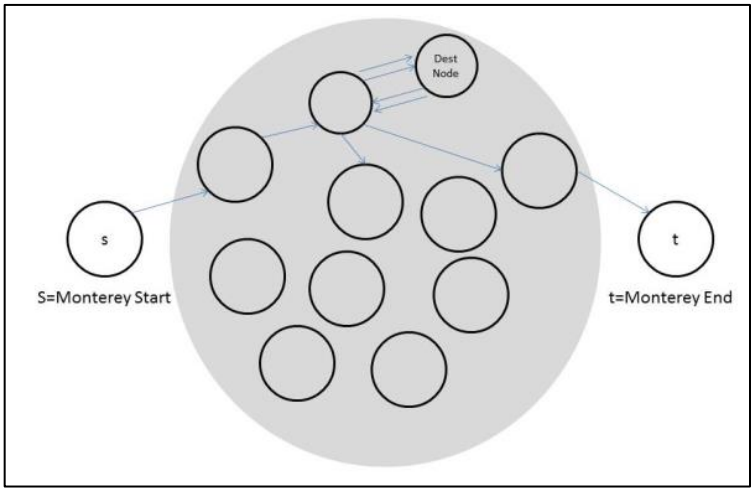


Figure 1: Network Overlay of San Francisco Bay Area

Choosing an Algorithm and Layering the Network

In order to capture time spent at each destination, dummy nodes were created for each destination. See Figure 2 for a conceptual picture. The time spent at those destinations is captured in the same manner as we capture the time to travel along any edge in the network. We add a 'fun' variable that is akin to a cost in the shortest path algorithm. 'Fun' is defined as the index (1 to 10) of the amount of fun gained from visiting a destination. We add this as a negative value to the edges to the dummy nodes, and through a layering process the entire undirected cyclic graph is converted into a directed acyclic graph. The result is that the shortest path algorithm produces results that provide the maximum fun.

To account for the time layered structure, a Python script was developed. The number of layers is a function the total trip length with units equal to the time unit desired. We choose to layer for every two minutes and round the time on network edges using floor division. For an idea of scale, 0830 to 2400 creates 48002 nodes and 109994 edges. A conceptual image of this technique is provided below in Figure 3.



↑ **Figure 2: Conceptual Image of adding dummy nodes to create a way to track stops at destination (Dst) nodes.**

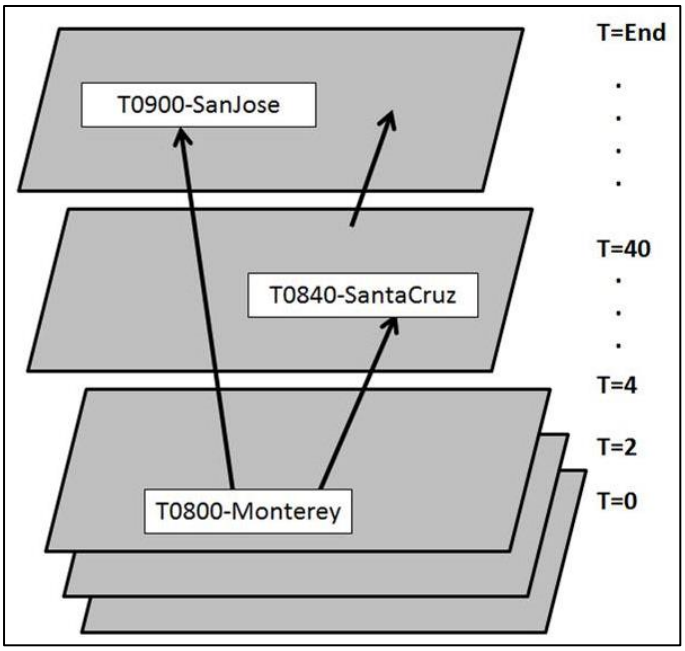


Figure 3: Conceptual Image of Time Layering showing how the time stamp creates a large directed acyclic graph (DAG).

Creating Interdictions

We simulate the concept of road construction (or other delays) by manually changing the delay factor for the affected arc in the file that is provided as an input to our python script. This method demonstrates the effect of road construction delays on our choice of destinations and ultimately our total 'fun'. As we increase this value, the model will update the optimal solution.

Adding Realism to the Problem

Rush hour traffic congestion is simulated in the python script by altering the time on arcs according to pre-defined scaling variables. This creates arcs that have longer lengths for the portion of time they are crossing time layers that correspond to pre-defined rush hours.

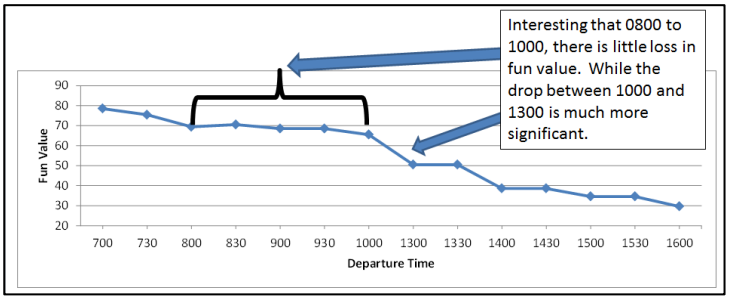
Repeat visits to destinations are given diminishing returns on 'fun' in the python script until no more 'fun' is achieved. Currently the max hours, at any destination, is limited to ten. The users control the initial value for 'fun' at each destination when entering their preferences.

Python script performs conditional checks when creating the layered network in order to tailor the arcs to appropriate times to visit each destination. For example, Muir Woods may open early, but visiting it at 10:00 p.m. is not realistic.

To prevent unnecessary time spent on the roads themselves (wandering), positive 'fun' values are assigned to all the road arcs based on time spent. The initial values provided in the input file to python are a simple formula: (Distance/Speed) x 0.1. This can easily be updated to accommodate different preferences.

Model Validation and Analysis of Results

Every trial produces a valid route, however to accurately demonstrate the product's usefulness the program is run with the same preferences but varying start times. Figures 4 and 5 illustrate the results from one such analysis. Analysis of other sets of preferences yield equally valid results, inclusive of route to be taken.



← **Figure 4: Plot illustrating how fun decreases as departure time increase, keeping return time steady at 2400.**

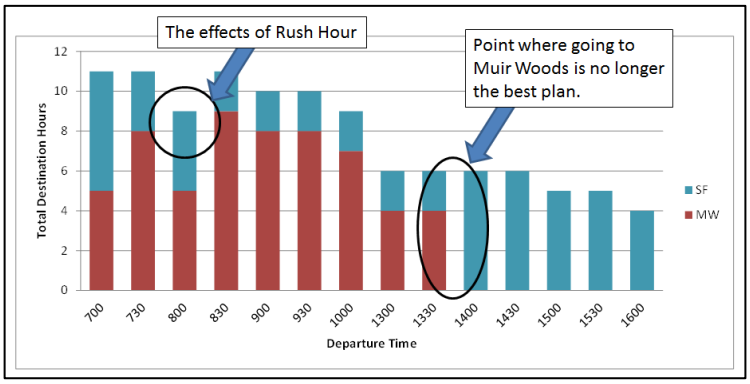


Figure 5: Plot illustrating total hours per scenario and the hours spent at each respective location as recommended.

Future Development of Additional Capabilities

Given more time, resources, and money we can improve the user interface to include a fully automated tool that requires the user to have no working knowledge of the software and other processes running in the background. Additional preference options can be added such as selecting start and end destinations. We can also improve the format of the output to create a finished report format. Included in that new output we can incorporate the ability to read batch files of more than one scenario and generate output that summarizes the recommendation much like we see in Figures 4 and 5. There is also potential to develop the one-day trip into a weekend, long weekend, or trip of customizable length. The network can also be expanded to include more destinations and expand beyond the San Francisco Bay Area.

Summary

This project ultimately answers two questions: (1) How much 'fun' can I have in a given day in the Bay Area? (2) What is the best route to take to achieve that level of 'fun'? The model as currently written can achieve these results and more.

We hope the B.A.D.A.S.S. Tool is useful to all of its users and becomes a regular point of consultation before deciding where to go and how to get there when there is no class on Friday!