Improving Emergency Vehicle Routing with Additional Road Features

Danny Brady[1] and Byungkyu Brian Park[2]
[1]*Data Science Insitute, University of Virginia, Charlottesville, VA 22904, U.S.A.*
[2]*Department of Civil and Environmental Engineering, University of Virginia, Charlottesville, VA 22904, U.S.A.*
{djb4t, bpark}@virginia.edu

Keywords: Emergency Vehicle Routing, Shortest Path, Dijkstra Algorithm, Road Features.

Abstract: Routing decisions affect everybody traveling on the roadways, but perhaps those decisions are never more important than for emergency responders. In a field where seconds count, getting lost or taking a sub-optimal route can mean the difference between life and death. Routing methods have advanced from hand drawn maps to smart phones with dynamic routing using GPS, and new data science methods can improve it further. This paper employs a modification to a common routing algorithm to improve route decision for emergency responders. By considering roadway geometry and connected vehicle technologies that simple routing algorithms ignore, Dijkstra's algorithm was able to be customized specifically for emergency response. The lane count, intersection control devices, and median construction were three additional elements of data incorporated into the modified model. Through a case study of an area of Albemarle County, Virginia, USA and discussion with domain experts, this paper demonstrates the feasibility and potential benefits of using such an algorithm to improve routing decisions for emergency responders.

1 INTRODUCTION

Emergency vehicle response technology has not been changed significantly over the last decade. Ten years ago, the fire service used paper map books to look up addresses, and then it was up to the individual responding to figure out the best route to the emergency incident. Today it is not much different, except that an address is typed in to software on a laptop which shows the location on a map on the computer. Firefighters are required to use their best discretion, local area knowledge and a bit of guesswork to determine how best to get to an incident. With the current level of mobile computing power, we propose that its capabilities should be harnessed to aid in determining the best driving route to emergency incidents.

There are numerous routing systems available to the average consumer, from built-in vehicle navigation to small dash-mounted units and personal smart phones. While these systems are useful in aiding transportation, there is additional information that these systems do not take into account that could prove extremely valuable to emergency vehicle operators. These additional features allow the creation of a routing system specifically tailored to emergency vehicle response which should aid the in the speed and safety of their travel. This paper proposed to utilize additional features on the transportation network such as the number of lanes, intersection control devices, and median construction, and to demonstrate the proposed shortest path finding algorithm modified from Dijkstra's algorithm is suitable for emergency vehicle route guidance.

2 METHODOLOGY

Standard route guidance systems use a relatively small number of features for their road network. Simple systems have information on link length and speeds, and their connections to other links. Since road geometry data is not easily accessible or accurate for freely online, we use tools like Google Maps to create an accurate representation of the road network served by Albemarle County's Seminole Trail Fire Station, in Virginia, USA. The novelty of this system is the various other road geometry features and Connected Vehicle technologies taken into account than simple systems are capable of analyzing.

Brady, D. and Park, B.
Improving Emergency Vehicle Routing with Additional Road Features.
In *Proceedings of the International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2016)*, pages 187-194
ISBN: 978-989-758-185-4

Included are features such as road medians that other vehicles can use to move out of the way and intersection pre-emption systems. Details of the road network used are in the Custom Road Network section.

The route guidance algorithm was implemented using a Python program and Javascript was used for visualizations. Dijkstra's algorithm, published in 1959, is a standard way to find the shortest distance on a graph. It utilizes a link weighting that determines the shortest "distance", and if that weight is simply the distance between road intersections the true shortest distance route can be found. However, we have created a novel weighting, or cost, by using the additional road network features so that the route chosen is not necessarily the shortest distance, but the most efficient for emergency vehicles. In order to determine how these additional features should affect the costs, we have used our first-hand knowledge of emergency responses as well as other domain experts in the area.

2.1 Algorithm Implementation

For the route guidance system, Dijkstra's algorithm was implemented into a Python code developed in this paper. The algorithm was originally created to find the shortest number of nodes between two points on a graph, but has been modified to consider the "cost" of each leg. This modification is key, because it allows the algorithm to optimize the path for various factors, not only distance. The algorithm reads in the road network data developed in this paper as described in the Custom Road Network section, and uses any cost functions, also implemented in Python. The algorithm can be seen in the Python Code section in Appendices, and the use of cost functions is discussed in more depth in the Cost Functions section.

2.2 Configuration

A vital step in the creation of a model or simulation is configuration. Configuration is the process of testing and adjusting a model to make it to match real world scenarios. To initially configure the algorithm and road network, a cost function that only used distances and road speeds was employed. When used in conjunction with Dijkstra's algorithm, this network yields results that mimic any other standard route guidance system. We tested several routes and was able to confirm that the routes generated using this simple cost function were typical shortest routes.

2.3 Road Network Visualization

Another important aspect of model creation is the use of visualizations. When creating a custom road network, a method to check for errors is to create a map. We tested this using the road network data we developed in this paper along with a simple webpage we created. The result is a map of the major roads in Albemarle County around the 29 North corridor. An open source javascript library called D3.js was used to layout our map as shown in Figure 1.

The lengths of each link represent the actual length between the indicated intersections. The color of each intersection represents the type of intersection control device there, either a stop sign (orange), a traffic light (light blue), or a traffic light with emergency vehicle pre-emption technology (dark blue). The thickness of each link also becomes greater as the speed limit increases. Using this map to help configure our network was very useful, as it is easy to see prominent area features. For example, the lines are thickest on 29, which has a speed limit of 45 mph. Those intersections also have traffic light pre-emption, while the smaller streets located in neighborhoods tend to have stop signs.

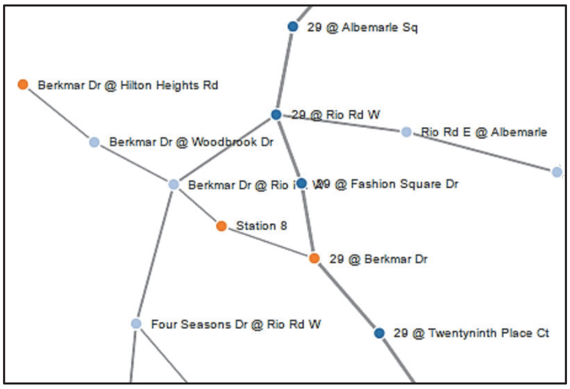

Figure 1: Roadway Network in Albemarle County.

2.4 Custom Road Network

There are is no open-source road network information available that is adequate for this paper. While there are freely available collections of road network data, they are currently not well suited for use in route guidance. It was determined that it would be satisfactory for the purposes of this paper to create custom road network data. Maps of the 29 North corridor were used as a base to determine which roads to include in the custom network. Smaller roads, especially dead ends are excluded because no matter the route taken to get to the smaller road, there is typically only one route to the final destination from there. This has the added benefit of keeping the network simpler and needing less custom data.

The road network data was stored using JavaScript Object Notation (JSON). It contains two main sections, the intersections and the roads of the network. Figure 2 shows examples from the actual network and the format used. The network contains all of the standard routing information including names of roads, their connections, their lengths in feet and their speed limits. Google maps distance measuring tool and street view were vital in gathering this information.

Example Intersections:

{"name":"29 @ Rio Rd","controlType":"preemptionSignal"}
{"name":"Station 8","controlType":"stop"}

Example Road Link:

{"source":"29 @ Hilton Heights Rd", "target":"29 @ Polo Grounds Rd", "speed":45, "distance":2400, "lanes":8, "median":"hard"}

Figure 2: Road Network Structure.

Additional features were required in order to improve route guidance for the specific use case of emergency responses. These features play a role in emergency response while not being very important for a typical driver. Information on the intersection control device, the number of total lanes on a link, as well as the median construction were added. The intersection control device was labeled as either a stop sign, traffic signal, or a pre-emption signal. Emergency vehicles could be equipped with Connected Vehicle technology that signals the infrastructure to change the light to green in the direction of their approach. These devices are general only installed on major roads as they can be costly to install and maintain, and hold less value on smaller roads with less vehicle volume.

The number of lanes does not normally play a role in routing for typical vehicles but is important for emergency vehicles. When an emergency vehicle is approaching, other drivers must move out of the way as quickly and carefully as possible. A road with eight lanes, such as Route 29, has more area for the vehicles to move to than a two lane road. It is therefore preferable for an emergency vehicle to travel on roads with more lanes if possible. Similarly, soft medians allow for other drivers to move out of the way even on a smaller road. This again is preferable to hard or non-existent medians that would involve a vehicle driving up over a curb of some kind.

2.5 Cost Functions

The first cost function created calibrated the route guidance system using the road's length to calculating a route's cost. This results in the true shortest path, but is more simplified than standard commercial route guidance algorithms. Using this cost function, we were able to quickly confirm that the code developed in this paper was working properly and that configuration was completed.

The second cost function implemented was one used in a commercial route guidance system. This implementation not only uses the road length, but weights it using the speed limit. This function should give the path with the shortest travel time under normal conditions. Some of the newer and more advanced software, such as Google Maps Directions and Waze, even account for current traffic conditions.

The last and most important cost function created was to optimize emergency vehicle response. This function builds off of the shortest travel time route, and adds additional factors for analysis. This function was modified by adding weighting factors based on each road's and intersection's unique features. In order to determine appropriate weight factors, and to assess practical implementation, this model was used in a case study of Seminole Trail Volunteer Fire Department (STVFD) response area. Domain experts from STVFD along with some trial and error were used to create these factors.

3 CASE STUDY

Albemarle County and the city of Charlottesville provide an interesting area for this case study. There are various types of roads and intersections with a mix of features and technologies ranging from small two-lane rural highways such as Barracks Road to huge intersections like Rio Rd at Route 29 with a total of 24 lane approaches. This variety provides a unique road network to test differences between standard and improved routing algorithms. In order to customize the proposed approach to Albemarle County, we spoke with several domain experts from Seminole Trail Volunteer Fire Department (STVFD) that all drive emergency vehicles on a regular basis. Their insight was vital to coming up with the extra road features considered most important for emergency responses. These experts described number of lanes, type of control device, and median construction as the most important factors. The overwhelming consensus was that the number of lanes is the most important factor. To take this into account, the cost function divides the cost of a link by the number of lanes, indicating that a 4 lane road is twice as preferable to a 2 lane road when everything else is equal.

In Albemarle County, the fire engines and ambulances are equipped with a device called an “Opticom.” These are lights mounted on the front of each vehicle that flash in a specific pattern or frequency, and that light is picked up by receivers mounted on or near several traffic signals on the most major roadways. These work with actuated signal controls to turn the proper approach green before the emergency vehicle arrives, leading to much easier traversal of the intersection by allowing for other vehicles to move through the intersection more easily than if the light was red.

When a route contained an intersection with a signal equipped with this technology, the cost of the link is lowered by 40%, a smaller amount than the lane count. Intersections with standard traffic signals were also reduced, but only by 20%. These intersections have green traffic lights some of the time when an emergency vehicle is approaching without pre-emption, so they are still preferable to a stop sign intersection. Since emergency vehicles should always stop at stop sign intersections until clear to proceed, these intersections received no cost reduction.

The least important factor is the median construction. From experience driving fire trucks, the average driver tends to attempt to move out of the way to the right, and not the left, completely ignoring the median. Still, if there is a soft median or even a two-way left turn lane in the middle of the road, it is typically better than a hard median or no median at all. If other vehicles try to move to the right as far as they can but there is no shoulder, these soft medians can be used by the emergency vehicle to get around more easily. The median construction type received the smallest cost adjustment in this system. The domain experts agreed that the least preferable type of median was a hard median, because even though most emergency vehicles could drive up over a curb, it generally is not done and could damage the vehicle. Hard medians received no cost reduction in this system. On many roads like two lane rural highways or smaller city roads, there is no median between the travel lanes, which was the second best option, receiving a cost reduction of 10%. While an emergency vehicle does not have to drive over a curb, they would have to enter an oncoming traffic lane which can be dangerous when done improperly. The best median construction was a soft median or two-way turn lane in the middle, because they provide plenty of additional space to move around other vehicles, and thus this type received the greatest cost reduction of 20%.

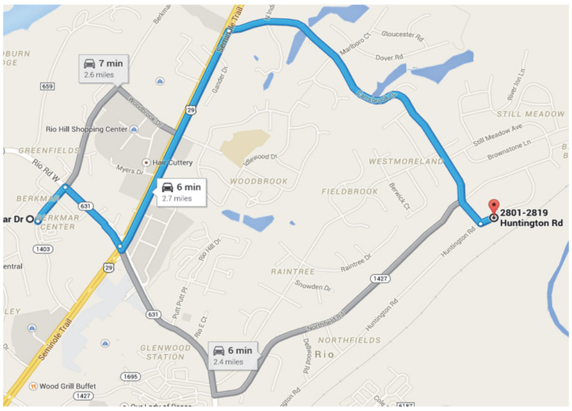

Figure 3: Demonstration of the proposed approach.

After coming up with the adjustment factors, we were able to demonstrate the proposed route guidance system to the experts by comparing the standard shortest distance algorithm to the proposed emergency route guidance system. As expected, several routes were the same with either method. This was the case when there was only one obvious route, such as from Route 29 and Hydraulic Road to East Rio Road and Albemarle Square. However, we were able to find multiple destinations that led to different routes being suggested. One example of this, as shown in Figure 3, was when determining the optimal route from STVFD on Berkmar Dr., between Route 29 and West Rio Road, to Carrsbrook Drive and Huntington Road. The shortest distance model suggests taking Northfield Rd. from East Rio Rd. for a distance of 2.4 miles. That route has several traffic lights and only one with the pre-emption technology, while the route the proposed approach suggests traveling north on Route 29 to Carrsbrook Road is longer at 2.7 miles, but has larger roads, a faster speed limit, and several lights with the pre-emption technology installed. This is the same route the experts regularly take instead of the shortest distance route.

4 CONCLUDING REMARKS

This paper proposed and demonstrated a route guidance approach for emergency vehicles. A Python based emergency vehicle route guidance algorithm code was developed and a JavaScript Object Notation

(JSON) was used to visualize the network used in this paper. It is noted that the Python code and JSON can be used with any network and cost functions as deemed by emergency vehicle domain experts.

While we could have applied recent advanced route guidance algorithms such as A* algorithm (Zeng and Church, 2009), dynamic traffic assignment-based guidance (Lee and Park, 2008), reliable route guidance (Nie et al., 2012), the main purpose of this paper was to demonstrate emergency vehicle route guidance by utilizing additional road features. Future research should consider implementing these advanced algorithms as well as generalized cost function instead of simple weighted average used in this paper.

In addition, technology is constantly making advances, and the field of traffic operations is no different. Emergency responders, and the fire service in particular, tend to hold onto tradition sometimes to their own detriment. There is technology available that, if adopted, could improve the efficiency of their responses with very little cost. With even relatively few additional pieces of information about the road network in an area, routing can be greatly improved. For a field in which seconds matter, personnel should strive for small improvements that could have a large impact, such as this incremental improvement over standard routing algorithms to improve speed and safety in responding to emergencies.

ACKNOWLEDGEMENTS

This research was in part supported by the Global Research Laboratory Program through the NRF funded by the Ministry of Science, ICT & Future Planning (2013K1A1A2A02078326). We are grateful for anonymous reviewers for their invaluable comments.

REFERENCES

Bray, Tim. "JSON Redux AKA RFC7159". Ongoing. Retrieved 25 November 2015.

Data-Driven Documents. n.d. from http://d3js.org.

Dijkstra, E. W. (1959). "A note on two problems in connexion with graphs". Numerische Mathematik 1: 269–271. doi:10.1007/BF01386390.

Google Maps. n.d. from http://www.maps.google.com.

Lee, J., Park, B., 2008. Evaluation of Route Guidance Strategies Based on Vehicle-Infrastructure Integration Under Incident Conditions. *Transportation Research Record: Journal of the Transportation Research Board 2086*, 107–114.

Nie, Y., Wu, X., Dillenburg, J.F., Nelson, P.C., 2012. Reliable route guidance: *A case study from Chicago. Transportation Research Part A*: Policy and Practice 46, 403–419.

Zeng, W., Church, R.L., 2009. Finding shortest paths on real road networks: the case for A*. *International Journal of Geographical Information Science 23*, 531–543.

APPENDIX

1 Python Code

```
__author__ = 'Danny Brady & B. Brian Park'
import json

    MEDIAN_ADJUSTMENT_FACTOR = {}
    MEDIAN_ADJUSTMENT_FACTOR['soft'] = 0.8
    MEDIAN_ADJUSTMENT_FACTOR['no'] = 0.9
    MEDIAN_ADJUSTMENT_FACTOR['hard'] = 1

    CONTROL_DEVICE_ADJUSTMENT_FACTOR = {}
    CONTROL_DEVICE_ADJUSTMENT_FACTOR['preempti
onSignal'] = 0.6
    CONTROL_DEVICE_ADJUSTMENT_FACTOR['signal'] =
0.8
    CONTROL_DEVICE_ADJUSTMENT_FACTOR['stop'] = 1

    def Dijkstra(graph, source, costFunction):
      unvisited_intersections = []
      dist = {}
      prev = {}
      dist[source] = 0
      for v in graph['intersections']:
        n = v['name']
        if n != source:
          dist[n] = float("inf")
          prev[n] = None
        unvisited_intersections.append(v)

      while len(unvisited_intersections) > 0:
        u = None
        current_shotest_distance = float('inf')
        for i in unvisited_intersections:
          if dist[i['name']] < current_shotest_distance:
            u = i
            current_shotest_distance = dist[i['name']]
        unvisited_intersections.remove(u)

        for neighbor_of_u in [road for road in graph['roads'] if
road['source'] == u['name'] or road['target'] == u['name']]:
          neighbor_name = neighbor_of_u['target'] if
neighbor_of_u['target'] != u['name'] else neighbor_of_u['source']
          neighbor = intersectionFromName(neighbor_name,
graph)
          alt = dist[u['name']] + costFunction(u['name'],
```

```
neighbor['name'], graph)
                if alt < dist[neighbor['name']]:
                    dist[neighbor['name']] = alt
                    prev[neighbor['name']] = u
        return (dist, prev)

    def intersectionFromName(name, graph):
        return [i for i in graph['intersections'] if i['name'] ==
name][0]

### Cost Functions ###

    def linkCostDistance(source, destination, graph):
        partial = [road for road in graph['roads'] if road['source']
== source or road['target'] == source]
        link = [road for road in partial if road['source'] ==
destination or road['target'] == destination][0]
        return float(link['distance'])


    def linkCostSpeedWeighted(source, destination, graph):
        partial = [road for road in graph['roads'] if road['source']
== source or road['target'] == source]
        link = [road for road in partial if road['source'] ==
destination or road['target'] == destination][0]
        return float(link['distance']) / float(link['speed'])

    def linkCostEmergency(source, destination, graph):
        partial = [road for road in graph['roads'] if road['source']
== source or road['target'] == source]
        link = [road for road in partial if road['source'] ==
destination or road['target'] == destination][0]
        cost = float(link['distance']) / float(link['speed']) * (8 /
float(link['lanes'])) *
MEDIAN_ADJUSTMENT_FACTOR[link['median']]
        source_intersection = intersectionFromName(source,
graph)
        destination_intersection =
intersectionFromName(destination, graph)
        cost = cost *
CONTROL_DEVICE_ADJUSTMENT_FACTOR[source_interse
ction['controlType']] * \

CONTROL_DEVICE_ADJUSTMENT_FACTOR[destination_in
tersection['controlType']]
        return cost

    if __name__ == "__main__":
        g = None
        with open("RoadNetwork.json") as json_file:
            g = json.load(json_file)
        #dist, prev = Dijkstra(g, "Station 8", linkCostDistance)
        #dist, prev = Dijkstra(g, "Station 8",
linkCostSpeedWeighted)
        dist, prev = Dijkstra(g, "Station 8", linkCostEmergency)
        dest_intersection = "Greenbrier Dr @ Whitewood Rd"
        #dest_intersection = "Berkmar Dr @ Hilton Heights Rd"
        path = []
        path.append(intersectionFromName(dest_intersection, g))
        while dest_intersection != "Station 8":
            p = prev[dest_intersection]
            path.append(p)
            dest_intersection = p['name']
        path.reverse()
        for p in path:
            print p
```

2 Complete Road Network Data


```
{
"intersections":[
        {"name":"29 @ Rio Rd
W","controlType":"preemptionSignal"},
        {"name":"Berkmar Dr @ Rio Rd
W","controlType":"signal"},
        {"name":"29 @ Albemarle
Sq","controlType":"preemptionSignal"},
        {"name":"29 @ Woodbrook
Dr","controlType":"preemptionSignal"},
        {"name":"29 @ Gander
Dr","controlType":"preemptionSignal"},
        {"name":"29 @ Hilton Heights
Rd","controlType":"preemptionSignal"},
        {"name":"29 @ Polo Grounds
Rd","controlType":"preemptionSignal"},
        {"name":"Four Seasons Dr @ Rio Rd
W","controlType":"signal"},
        {"name":"Four Seasons Dr @ Commonwealth
Dr","controlType":"stop"},
        {"name":"Commonwealth Dr @ Westfield
Rd","controlType":"stop"},
        {"name":"Commonwealth Dr @ Greenbrier
Dr","controlType":"stop"},
        {"name":"Earlysville Rd @ Rio Rd
W","controlType":"signal"},
        {"name":"Hydraulic Rd @ Whitewood
Rd","controlType":"signal"},
        {"name":"Greenbrier Dr @ Whitewood
Rd","controlType":"stop"},
        {"name":"Hydraulic Rd @ Georgetown
Rd","controlType":"signal"},
        {"name":"Barracks Rd @ Georgetown
Rd","controlType":"signal"},
        {"name":"Hydraulic Rd @ Commonwealth
Dr","controlType":"signal"},
        {"name":"Hydraulic Rd @ District
Ave","controlType":"signal"},
        {"name":"29 @ Hydraulic
Rd","controlType":"signal"},
        {"name":"29 @ Fashion Square
Dr","controlType":"preemptionSignal"},
        {"name":"29 @ Twentyninth Place
Ct","controlType":"preemptionSignal"},
        {"name":"29 @ Branchlands
```

Blvd","controlType":"preemptionSignal"},
{"name":"29 @ Greenbrier Dr","controlType":"preemptionSignal"},
{"name":"29 @ Seminol Ct","controlType":"preemptionSignal"},
{"name":"Berkmar Dr @ Woodbrook Dr","controlType":"signal"},
{"name":"Berkmar Dr @ Hilton Heights Rd","controlType":"stop"},
{"name":"Rio Rd E @ Albemarle Square","controlType":"signal"},
{"name":"Rio Rd E @ Old Brook Rd","controlType":"signal"},
{"name":"Rio Rd E @ Hillsdale Dr","controlType":"signal"},
{"name":"Rio Rd E @ Huntington Rd","controlType":"signal"},
{"name":"Rio Rd E @ Greenbrier Dr","controlType":"signal"},
{"name":"Rio Rd E @ Meadowcreek Parkway","controlType":"signal"},
{"name":"Rio Rd E @ Melbourne Rd","controlType":"signal"},
{"name":"Station 8","controlType":"stop"},
{"name":"29 @ Carrsbrook Dr","controlType":"stop"},
{"name":"Carrsbrook Dr @ Old Brook Rd","controlType":"stop"},
{"name":"Carrsbrook Dr @ Hillsdale Dr","controlType":"stop"},
{"name":"Carrsbrook Dr @ Huntington Rd","controlType":"stop"},
{"name":"Meadowcreek Parkway @ Melbourne Rd","controlType":"stop"},
{"name":"Branchlands Blvd @ Hillsdale Dr","controlType":"stop"},
{"name":"Hillsdale Dr @ Greenbrier Dr","controlType":"stop"},
{"name":"29 @ Berkmar Dr","controlType":"stop"}
],

"roads":[
{"source":"29 @ Hilton Heights Rd", "target":"29 @ Polo Grounds Rd", "speed":45, "distance":2400, "lanes":8, "median":"hard"},
{"source":"29 @ Hilton Heights Rd", "target":"29 @ Carrsbrook Dr", "speed":45, "distance":1465, "lanes":8, "median":"hard"},
{"source":"29 @ Gander Dr", "target":"29 @ Carrsbrook Dr", "speed":45, "distance":610, "lanes":8, "median":"hard"},
{"source":"29 @ Woodbrook Dr", "target":"29 @ Gander Dr", "speed":45, "distance":1650, "lanes":8, "median":"hard"},
{"source":"29 @ Woodbrook Dr", "target":"29 @ Albemarle Sq", "speed":45, "distance":1685, "lanes":8, "median":"hard"},
{"source":"29 @ Rio Rd W", "target":"29 @ Albemarle Sq", "speed":45, "distance":685, "lanes":8, "median":"hard"},
{"source":"29 @ Rio Rd W", "target":"29 @ Fashion Square Dr", "speed":45, "distance":980, "lanes":8, "median":"hard"},
{"source":"29 @ Berkmar Dr", "target":"29 @ Fashion Square Dr", "speed":45, "distance":520, "lanes":8, "median":"hard"},
{"source":"29 @ Berkmar Dr", "target":"29 @ Twentyninth Place Ct", "speed":45, "distance":750, "lanes":8, "median":"hard"},
{"source":"29 @ Twentyninth Place Ct", "target":"29 @ Branchlands Blvd", "speed":45, "distance":1210, "lanes":8, "median":"hard"},
{"source":"29 @ Greenbrier Dr", "target":"29 @ Branchlands Blvd", "speed":45, "distance":1750, "lanes":8, "median":"hard"},
{"source":"29 @ Greenbrier Dr", "target":"29 @ Seminol Ct", "speed":45, "distance":1910, "lanes":8, "median":"hard"},
{"source":"29 @ Hydraulic Rd", "target":"29 @ Seminol Ct", "speed":45, "distance":1680, "lanes":8, "median":"hard"},
{"source":"29 @ Rio Rd W", "target":"Berkmar Dr @ Rio Rd W", "speed":35, "distance":1485, "lanes":4, "median":"no"},
{"source":"Four Seasons Dr @ Rio Rd W", "target":"Berkmar Dr @ Rio Rd W", "speed":35, "distance":3255, "lanes":4, "median":"no"},
{"source":"Four Seasons Dr @ Rio Rd W", "target":"Earlysville Rd @ Rio Rd W", "speed":35, "distance":1260, "lanes":4, "median":"no"},
{"source":"Hydraulic Rd @ Whitewood Rd", "target":"Earlysville Rd @ Rio Rd W", "speed":35, "distance":3455, "lanes":4, "median":"no"},
{"source":"Hydraulic Rd @ Whitewood Rd", "target":"Hydraulic Rd @ Georgetown Rd", "speed":35, "distance":1620, "lanes":4, "median":"no"},
{"source":"Barracks Rd @ Georgetown Rd", "target":"Hydraulic Rd @ Georgetown Rd", "speed":35, "distance":4525, "lanes":2, "median":"no"},
{"source":"Hydraulic Rd @ Commonwealth Dr", "target":"Hydraulic Rd @ Georgetown Rd", "speed":35, "distance":1815, "lanes":4, "median":"no"},
{"source":"Hydraulic Rd @ Commonwealth Dr", "target":"Hydraulic Rd @ District Ave", "speed":35, "distance":950, "lanes":4, "median":"no"},
{"source":"Hydraulic Rd @ Commonwealth Dr", "target":"Commonwealth Dr @ Greenbrier Dr", "speed":35, "distance":3165, "lanes":2, "median":"soft"},
{"source":"29 @ Hydraulic Rd", "target":"Hydraulic Rd @ District Ave", "speed":35, "distance":1185, "lanes":4, "median":"no"},
{"source":"Berkmar Dr @ Rio Rd W", "target":"Station 8", "speed":35, "distance":880, "lanes":2,

"median":"no"},
{"source":"Station 8", "target":"29 @ Berkmar Dr", "speed":35, "distance":1920, "lanes":2, "median":"no"},
{"source":"Berkmar Dr @ Rio Rd W", "target":"Berkmar Dr @ Woodbrook Dr", "speed":35, "distance":2180, "lanes":2, "median":"no"},
{"source":"Berkmar Dr @ Hilton Heights Rd", "target":"Berkmar Dr @ Woodbrook Dr", "speed":35, "distance":3660, "lanes":2, "median":"no"},
{"source":"29 @ Rio Rd W", "target":"Rio Rd E @ Albemarle Square", "speed":35, "distance":835, "lanes":4, "median":"no"},
{"source":"Rio Rd E @ Old Brook Rd", "target":"Rio Rd E @ Albemarle Square", "speed":35, "distance":2400, "lanes":4, "median":"no"},
{"source":"Rio Rd E @ Old Brook Rd", "target":"Rio Rd E @ Hillsdale Dr", "speed":35, "distance":210, "lanes":4, "median":"no"},
{"source":"Rio Rd E @ Huntington Rd", "target":"Rio Rd E @ Hillsdale Dr", "speed":35, "distance":2095, "lanes":4, "median":"soft"},
{"source":"Rio Rd E @ Huntington Rd", "target":"Rio Rd E @ Greenbrier Dr", "speed":35, "distance":670, "lanes":4, "median":"soft"},
{"source":"Rio Rd E @ Greenbrier Dr", "target":"Rio Rd E @ Meadowcreek Parkway", "speed":35, "distance":2475, "lanes":4, "median":"soft"},
{"source":"Rio Rd E @ Meadowcreek Parkway", "target":"Rio Rd E @ Melbourne Rd", "speed":35, "distance":7180, "lanes":2, "median":"no"},
{"source":"Rio Rd E @ Meadowcreek Parkway", "target":"Meadowcreek Parkway @ Melbourne Rd", "speed":35, "distance":5330, "lanes":2, "median":"no"},
{"source":"29 @ Greenbrier Dr", "target":"Hillsdale Dr @ Greenbrier Dr", "speed":25, "distance":1215, "lanes":2, "median":"soft"},
{"source":"Branchlands Blvd @ Hillsdale Dr", "target":"Hillsdale Dr @ Greenbrier Dr", "speed":35, "distance":1635, "lanes":2, "median":"soft"},
{"source":"Branchlands Blvd @ Hillsdale Dr", "target":"Rio Rd E @ Hillsdale Dr", "speed":35, "distance":3335, "lanes":2, "median":"soft"},
{"source":"29 @ Carrsbrook Dr", "target":"Carrsbrook Dr @ Old Brook Rd", "speed":25, "distance":5095, "lanes":2, "median":"no"},
{"source":"Carrsbrook Dr @ Hillsdale Dr", "target":"Carrsbrook Dr @ Old Brook Rd", "speed":25, "distance":1055, "lanes":2, "median":"no"},
{"source":"Carrsbrook Dr @ Hillsdale Dr", "target":"Carrsbrook Dr @ Huntington Rd", "speed":25, "distance":555, "lanes":2, "median":"no"},
{"source":"Rio Rd E @ Old Brook Rd", "target":"Carrsbrook Dr @ Old Brook Rd", "speed":30, "distance":6285, "lanes":2, "median":"no"},
{"source":"Rio Rd E @ Hillsdale Dr", "target":"Carrsbrook Dr @ Hillsdale Dr", "speed":30, "distance":6020, "lanes":2, "median":"no"},
{"source":"Rio Rd E @ Huntington Rd", "target":"Carrsbrook Dr @ Huntington Rd", "speed":30, "distance":7285, "lanes":2, "median":"no"},
{"source":"Commonwealth Dr @ Westfield Rd", "target":"Commonwealth Dr @ Greenbrier Dr", "speed":25, "distance":1055, "lanes":2, "median":"no"},
{"source":"Commonwealth Dr @ Westfield Rd", "target":"Four Seasons Dr @ Commonwealth Dr", "speed":25, "distance":1545, "lanes":2, "median":"no"},
{"source":"Four Seasons Dr @ Rio Rd W", "target":"Four Seasons Dr @ Commonwealth Dr", "speed":25, "distance":3515, "lanes":2, "median":"no"},

{"source":"Greenbrier Dr @ Whitewood Rd", "target":"Hydraulic Rd @ Whitewood Rd", "speed":25, "distance":2370, "lanes":2, "median":"no"},
{"source":"Greenbrier Dr @ Whitewood Rd", "target":"Commonwealth Dr @ Greenbrier Dr", "speed":25, "distance":510, "lanes":2, "median":"no"},

{"source":"29 @ Greenbrier Dr", "target":"Commonwealth Dr @ Greenbrier Dr", "speed":25, "distance":1725, "lanes":2, "median":"soft"},

{"source":"29 @ Woodbrook Dr", "target":"Berkmar Dr @ Woodbrook Dr", "speed":25, "distance":1390, "lanes":4, "median":"hard"},
{"source":"29 @ Hilton Heights Rd", "target":"Berkmar Dr @ Hilton Heights Rd", "speed":25, "distance":1070, "lanes":4, "median":"hard"}
]
}