Calculate Roadway Congestion

Purpose

Quantifying and communicating congestion on highways and local streets in Texas is challenging. A clear, replicable, and easily understandable method is required to help manage, score, and evaluate planned projects in the Transportation Planning and Programming Division.

This process will produce congestion data to display on maps in order to identify roads that are most impacted by congestion, communicate congestion to the public, and help the decision-making process by identifying projects where improvements can be made to provide relief.

Method

This process builds on work conducted in the Unified Transportation Program (UTP) Ranking and Scoring tool but differs by not using INRIX speed or daily volume to capacity ratios. This new method, referred to as the **"Car-Space" Method**, differs from other, more traditional methods, by determining the space between cars in one mile increments. This method factors number of lanes, Annual Average Daily Traffic (AADT), and average car length to calculate the space remaining between vehicles for each 1 mile segment of roadway during the 30th peak hour.

The 30th peak hour (or k-factor) is typically used in traffic analysis and represents the 30th highest hour of traffic counts for the entire year (365 days, every hour of every day) at that permanent station site. *(Source: TxDOT TPP Traffic Analysis)*

Setup

Software required to run process:

- ESRI ArcMap
- Python or any IDE (integrated development environment)
- Notepad or any text editor

Create the following directories:

- Project Directory – C:\TxDOT\Congestion
- Geodatabase – C:\TxDOT\Congestion\Data
    - Add TxDOT_Roadway_Inventory.gdb to folder
- Python Script – C:\TxDOT\Congestion\Scripts

    - Add a copy of the script, Congestion_Calculations.py, to folder

- Documentation – C:\TxDOT\Congestion\Process

Download TxDOT End of Year data:

http://www.txdot.gov/inside-txdot/division/transportation-planning/roadway-inventory.html

- Unzip the download to C:\TxDOT\Congestion\Data and Use the feature class layer file, **TxDOT_Roadway_Linework_Routed**, from the downloaded data.
- This process is designed for centerline data. The query RDBD_ID = 'CG' will provide centerline data, however to get On-System centerlines only, use REC = 1, which will exclude right/left roadbeds & include only On-System routes.
- Highway Status needs to be Open to Traffic, use HWY_STAT > 3.
- In the *Layer Properties* for **TxDOT_Roadway_Linework_Routed,** apply the filter, REC = 1 AND HWY_STAT > 3 under *Definition Query.*
- Export selected records as a feature class layer to **TxDOT_Roadway_Inventory.gdb** in the downloaded data.
    - Name the file, **TxDOT_Roadway_ Routed_OnSystem**.
    - The RDBD_ID asset in this file should only have the attribute CG for centerline.
    - This file should only have around 103,004 records. Although each year there will likely be a slight change depending on the number of new and removed on-system roads.

Note**:** *It is very important to* ***name the file generated exactly as stated*** *in this process. The Python script will not be able to find the file without the precise naming convention.*

This process can be applied to all roadway segments (on and off system) or a selection of routes by applying the appropriate definition query. Use the TxDOT_Roadway_Inventory.pdf file from the downloaded data to determine the appropriate query. This file describes all the assets available.

Procedures

Step 1 – Click on the link, Python Script (a.k.a. Congestion_Calculations.py) which is located toward the end of this document.

- Copy this script into a Python shell in ArcMap or any IDE (integrated development environment) application such as Pycharm or PyScripter or others.
- May want to save as Congestion_Calculations.py file for future use as well.
- Run script against the **TxDOT_Roadway_Routed_OnSystem** layer created earlier.

- Make sure the script points to where the layer is located.

Step 2 – After the Python script has finished in Step 1, new fields will be added to **TxDOT_Roadway_Routed_OnSystem**.

- The script used in **Step 1**, is an abbreviated version that adds 8 new fields and comprises of 2 scenarios, 4 fields each
    - Base Year (Current Year of TxDOT Roadways data used) (Based on ADT_CUR)
    - Forecast Year (20 years in the future from Current Year) (Based on ADT_DESGN)
- Currently, the Statewide Planning Map only has these first 2 scenarios
- The Full Python Script, at the end of this document, includes the 2 scenarios listed above along with 2 additional scenarios, 4 fields each, 16 total. Use this version instead, if desire other scenarios.
    - Base Year Alternative (ADT_CUR & increases number of lanes by a factor of 50%)
    - Forecast Year Alternative (ADT_DESGN & increases number of lanes by a factor of 50%)
- **Four fields** that are added for each scenario and the equation used to produce the data:
    - **AVG_CAR_LENGTH** = 15
    - **TRUCKS** = [ADT_CUR] * ([TRK_AADT_PCT] * .01)
    - **CARS_PER_MIN** = ((([ADT_CUR] + [TRK_AADT_PCT]) * ([K_FAC] *.01)) / [NUM_LANES]) / 60
    - **CAR_SPACE** = ((5280) - ((((([ADT_CUR] + [TRUCKS]) * ([K_FAC] *.01)) / [NUM_LANES]) / 60) * [AVG_CAR_LENGTH])) / [CARS_PER_MIN]

The factors for each scenario can be adjusted in the script to allow for scenario variations:

- Average Car Length
    - Script defaults to 15 feet. This number can be disputed, adjust script, in feet, as desired.
- Analysis Unit
    - Factor is normalized to 5,280 feet of roadway (1 mile), adjust script, in feet, as desired.
- Truck Factor
    - Adds trucks to the existing AADT. Default is .01 (Truck Factor of 2)
    - Truck Factor = ".015" (Truck Factor of 2.5)
    - Truck Factor = ".02" (Truck Factor of 3)
- Capacity Factor
    - Accounts for increased roadway capacity for alternatives from Base Year Calculation

- Default increase is 10%
- Capacity Factor = "1.1" (Increases roadway capacity by 10%
- Capacity Factor = "1.2" (Increases roadway capacity by 20%
- Capacity Factor = "1.3" (Increases roadway capacity by 30%
- Capacity Factor = "1.4" (Increases roadway capacity by 40%)
- Capacity Factor = "1.5" (Increases roadway capacity by 50%)

Step 3 – The end result of the script is the calculated space, in feet, remaining between cars. Categorize to illustrate levels of congestion using Graduated colors under the Layer Properties Symbology tab.

Symbolize the data using the CAR_SPACE column as follows:

- Congested, Car_Space < 175 (Red, 4 line width)
- Moderately Congested, Car_Space >= 175 and Car_Space < 350 (Orange, 2 line width)
- Not Congested, Car_Space >= 350 (Light gray, 0.1 line width)

Step 4 – Add layer to a map with TxDOT Roadways linework centerline, cities, counties and districts.

- Review the results in urban and rural areas for obvious errors or questionable results.
- Compare results to previous results with the congestion data on TxDOT's Open Data Portal
- After passing inspection, export to a PDF and plot map if desired, depending on need.

Python Script (*abbreviated version, 2 scenarios*) (Link back to Step 2)

```
import arcpy

#Roadway Inventory Dataset (MUST HAVE THE INVENTORY DATA FOR ADT_CUR, TRK_AADT_PCT, NUM_LANES, K_FAC)

TxDOT_Roadway_Linework_Routed = "C:\\TxDOT\\Congestion\\Data\\TxDOT_Roadway_Inventory.gdb\\TxDOT_Roadway_Routed_OnSystem"

#Average Car Length, default is 15

averageCarLength = "15"

#Analysis Unit, the calculation is normalized to a 5280 foot section of roadway

analysisUnit = "5280"
```

```
#Truck Factor adds trucks to the existing AADT (trucks are already in the AADT).  Default is .01

truckFactor = ".01"  #Truck Factor of 2

#truckFactor = ".015"  #Truck Factor of 2.5

#truckFactor = ".02"  #Truck Factor of 3

 #Increased Capacity for Alternative Calculations, default increase is %10

#capacityFactor = "1.1" #Increase roadway capacity by 10%

#capacityFactor = "1.2" #Increase roadway capacity by 20%

#capacityFactor = "1.3" #Increase roadway capacity by 30%

#capacityFactor = "1.4" #Increase roadway capacity by 40%

capacityFactor = "1.5" #Increase roadway capacity by 50%


#print "Adding Fields...."

arcpy.AddField_management(TxDOT_Roadway_Linework_Routed, "AVG_CAR_LENGTH", "FLOAT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")

arcpy.AddField_management(TxDOT_Roadway_Linework_Routed, "CARS_PER_MIN", "FLOAT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")

arcpy.AddField_management(TxDOT_Roadway_Linework_Routed, "CAR_SPACE", "FLOAT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")

arcpy.AddField_management(TxDOT_Roadway_Linework_Routed, "TRUCKS", "LONG", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")


print "Calculating Average Car Length...."

arcpy.CalculateField_management(TxDOT_Roadway_Linework_Routed, "AVG_CAR_LENGTH", averageCarLength, "VB", "")

print "Calculating Number of Trucks...."

arcpy.CalculateField_management(TxDOT_Roadway_Linework_Routed, "TRUCKS", "[ADT_CUR] * ([TRK_AADT_PCT] * " + truckFactor + ")", "VB", "")
```

```
print "Calculating Cars Per Minute Per Lane...."

arcpy.CalculateField_management(TxDOT_Roadway_Linework_Routed, "CARS_PER_MIN",
"((([ADT_CUR] + [TRUCKS]) * ( [K_FAC] *.01)) / [NUM_LANES]) / 60", "VB", "")



print "Calculating Car Spacing...."

arcpy.CalculateField_management(TxDOT_Roadway_Linework_Routed, "CAR_SPACE", "((" +
analysisUnit + ") - (((((([ADT_CUR] + [TRUCKS]) * ( [K_FAC] *.01)) / [NUM_LANES]) / 60) *
[AVG_CAR_LENGTH])) / [CARS_PER_MIN]", "VB", "")

#Forecast Year Calculations

print "Adding Fields For Forecast Year...."
arcpy.AddField_management(TxDOT_Roadway_Linework_Routed, "FORECAST_AVG_CAR_LENGTH",
"FLOAT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")
arcpy.AddField_management(TxDOT_Roadway_Linework_Routed, "FORECAST_CARS_PER_MIN",
"FLOAT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")
arcpy.AddField_management(TxDOT_Roadway_Linework_Routed, "FORECAST_CAR_SPACE", "FLOAT",
"", "", "", "", "NULLABLE", "NON_REQUIRED", "")
arcpy.AddField_management(TxDOT_Roadway_Linework_Routed, "FORECAST_TRUCKS", "LONG", "", "",
"", "", "NULLABLE", "NON_REQUIRED", "")

print "Adding Average Car Length...."
arcpy.CalculateField_management(TxDOT_Roadway_Linework_Routed,
"FORECAST_AVG_CAR_LENGTH", averageCarLength, "VB", "")

print "Calculating Number of Trucks...."
arcpy.CalculateField_management(TxDOT_Roadway_Linework_Routed, "FORECAST_TRUCKS",
"[ADT_DESGN] * ([TRK_AADT_PCT] * " + truckFactor + ")", "VB", "")

print "Calculating Cars Per Minute Per Lane...."
arcpy.CalculateField_management(TxDOT_Roadway_Linework_Routed, "FORECAST_CARS_PER_MIN",
"((([ADT_DESGN] + [FORECAST_TRUCKS]) * ( [K_FAC] *.01)) / [NUM_LANES]) / 60", "VB", "")

print "Calculating Car Spacing...."
arcpy.CalculateField_management(TxDOT_Roadway_Linework_Routed, "FORECAST_CAR_SPACE", "((" +
analysisUnit + ") - (((((([ADT_DESGN] + [FORECAST_TRUCKS]) * ( [K_FAC] *.01)) / [NUM_LANES]) / 60) *
[FORECAST_AVG_CAR_LENGTH])) / [FORECAST_CARS_PER_MIN]", "VB", "")

 print "Script Complete"
```

Full Python Script (*full version, 4 scenarios*)(Link back to Step 2)

```
import arcpy

#Roadway Inventory Dataset (MUST HAVE THE INVENTORY DATA FOR ADT_CUR, TRK_AADT_PCT,

NUM_LANES, K_FAC)

TxDOT_Roadway_Linework_Routed =
"C:\\TxDOT\\Congestion\\TxDOT_Roadways_Inventory.gdb\\TxDOT_Roadway_Routed_OnSystem"


#Average Car Length, default is 15

averageCarLength = "15"


#Analysis Unit, the calculation is normalized to a 5280 foot section of roadway

analysisUnit = "5280"


#Truck Factor adds trucks to the existing AADT (trucks are already in the AADT).  Default is .01

truckFactor = ".01"  #Truck Factor of 2

#truckFactor = ".015"  #Truck Factor of 2.5

#truckFactor = ".02"  #Truck Factor of 3


#Increased Capacity for Alternative Calculations, default increase is %10

#capacityFactor = "1.1" #Increase roadway capacity by 10%

#capacityFactor = "1.2" #Increase roadway capacity by 20%

#capacityFactor = "1.3" #Increase roadway capacity by 30%

#capacityFactor = "1.4" #Increase roadway capacity by 40%

capacityFactor = "1.5" #Increase roadway capacity by 50%
```

```
#

#Base Year Calculations

#

print "Adding Fields For Base Year...."

arcpy.AddField_management(TxDOT_Roadway_Linework_Routed, "BASE_AVG_CAR_LENGTH", "FLOAT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")

arcpy.AddField_management(TxDOT_Roadway_Linework_Routed, "BASE_CARS_PER_MIN", "FLOAT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")

arcpy.AddField_management(TxDOT_Roadway_Linework_Routed, "BASE_CAR_SPACE", "FLOAT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")

arcpy.AddField_management(TxDOT_Roadway_Linework_Routed, "BASE_TRUCKS", "LONG", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")


print "Adding Average Car Length...."

arcpy.CalculateField_management(TxDOT_Roadway_Linework_Routed, "BASE_AVG_CAR_LENGTH", averageCarLength, "VB", "")


print "Calculating Number of Trucks...."

arcpy.CalculateField_management(TxDOT_Roadway_Linework_Routed, "BASE_TRUCKS", "[ADT_CUR] * ([TRK_AADT_PCT] * " + truckFactor + ")", "VB", "")


print "Calculating Cars Per Minute Per Lane...."

arcpy.CalculateField_management(TxDOT_Roadway_Linework_Routed, "BASE_CARS_PER_MIN", "((([ADT_CUR] + [BASE_TRUCKS]) * ( [K_FAC] *.01)) / [NUM_LANES]) / 60", "VB", "")


print "Calculating Car Spacing...."
```

```
arcpy.CalculateField_management(TxDOT_Roadway_Linework_Routed, "BASE_CAR_SPACE", "((" + analysisUnit + ") - ((((([ADT_CUR] + [BASE_TRUCKS]) * ( [K_FAC] *.01)) / [NUM_LANES]) / 60) * [BASE_AVG_CAR_LENGTH])) / [BASE_CARS_PER_MIN]", "VB", "")

#

#Base Year with Alternative Calculations

#

print "Adding Fields For Base Year Improvements...."

arcpy.AddField_management(TxDOT_Roadway_Linework_Routed, "BASE_ALTERNATIVE_AVG_CAR_LENGTH", "FLOAT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")

arcpy.AddField_management(TxDOT_Roadway_Linework_Routed, "BASE_ALTERNATIVE_CARS_PER_MIN", "FLOAT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")

arcpy.AddField_management(TxDOT_Roadway_Linework_Routed, "BASE_ALTERNATIVE_CAR_SPACE", "FLOAT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")

arcpy.AddField_management(TxDOT_Roadway_Linework_Routed, "BASE_ALTERNATIVE_TRUCKS", "LONG", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")


print "Adding Average Car Length...."

arcpy.CalculateField_management(TxDOT_Roadway_Linework_Routed, "BASE_ALTERNATIVE_AVG_CAR_LENGTH", averageCarLength, "VB", "")


print "Calculating Number of Trucks...."

arcpy.CalculateField_management(TxDOT_Roadway_Linework_Routed, "BASE_ALTERNATIVE_TRUCKS", "[ADT_CUR] * ([TRK_AADT_PCT] * " + truckFactor + ")", "VB", "")


print "Calculating Cars Per Minute Per Lane...."

arcpy.CalculateField_management(TxDOT_Roadway_Linework_Routed, "BASE_ALTERNATIVE_CARS_PER_MIN", "((([ADT_CUR] + [BASE_ALTERNATIVE_TRUCKS]) * ( [K_FAC] *.01)) / ([NUM_LANES] * " + capacityFactor + ")) / 60", "VB", "")

print "Calculating Car Spacing...."
```

```
arcpy.CalculateField_management(TxDOT_Roadway_Linework_Routed, "BASE_ALTERNATIVE_CAR_SPACE", "((" + analysisUnit + ") - ((((([ADT_CUR] + [BASE_ALTERNATIVE_TRUCKS]) * ( [K_FAC] *.01)) / ([NUM_LANES] * " + capacityFactor + ")) / 60) * [BASE_ALTERNATIVE_AVG_CAR_LENGTH])) / [BASE_ALTERNATIVE_CARS_PER_MIN]", "VB", "")

#

#Forecast Year Calculations

#

print "Adding Fields For Forecast Year...."

arcpy.AddField_management(TxDOT_Roadway_Linework_Routed, "FORECAST_AVG_CAR_LENGTH", "FLOAT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")

arcpy.AddField_management(TxDOT_Roadway_Linework_Routed, "FORECAST_CARS_PER_MIN", "FLOAT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")

arcpy.AddField_management(TxDOT_Roadway_Linework_Routed, "FORECAST_CAR_SPACE", "FLOAT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")

arcpy.AddField_management(TxDOT_Roadway_Linework_Routed, "FORECAST_TRUCKS", "LONG", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")


print "Adding Average Car Length...."

arcpy.CalculateField_management(TxDOT_Roadway_Linework_Routed, "FORECAST_AVG_CAR_LENGTH", averageCarLength, "VB", "")


print "Calculating Number of Trucks...."

arcpy.CalculateField_management(TxDOT_Roadway_Linework_Routed, "FORECAST_TRUCKS", "[ADT_DESGN] * ([TRK_AADT_PCT] * " + truckFactor + ")", "VB", "")


print "Calculating Cars Per Minute Per Lane...."

arcpy.CalculateField_management(TxDOT_Roadway_Linework_Routed, "FORECAST_CARS_PER_MIN", "((([ADT_DESGN] + [FORECAST_TRUCKS]) * ( [K_FAC] *.01)) / [NUM_LANES]) / 60", "VB", "")


print "Calculating Car Spacing...."
```

```
arcpy.CalculateField_management(TxDOT_Roadway_Linework_Routed, "FORECAST_CAR_SPACE", "((" + analysisUnit + ") - ((((([ADT_DESGN] + [FORECAST_TRUCKS]) * ( [K_FAC] *.01)) / [NUM_LANES]) / 60) * [FORECAST_AVG_CAR_LENGTH])) / [FORECAST_CARS_PER_MIN]", "VB", "")

#

#Forecast Year with Alternative Calculations

#

print "Adding Fields For Forecast Year...."

arcpy.AddField_management(TxDOT_Roadway_Linework_Routed, "FORECAST_ALTERNATIVE_AVG_CAR_LENGTH", "FLOAT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")

arcpy.AddField_management(TxDOT_Roadway_Linework_Routed, "FORECAST_ALTERNATIVE_CARS_PER_MIN", "FLOAT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")

arcpy.AddField_management(TxDOT_Roadway_Linework_Routed, "FORECAST_ALTERNATIVE_CAR_SPACE", "FLOAT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")

arcpy.AddField_management(TxDOT_Roadway_Linework_Routed, "FORECAST_ALTERNATIVE_TRUCKS", "LONG", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")


print "Adding Average Car Length...."

arcpy.CalculateField_management(TxDOT_Roadway_Linework_Routed, "FORECAST_ALTERNATIVE_AVG_CAR_LENGTH", averageCarLength, "VB", "")

print "Calculating Number of Trucks...."

arcpy.CalculateField_management(TxDOT_Roadway_Linework_Routed, "FORECAST_ALTERNATIVE_TRUCKS", "[ADT_DESGN] * ([TRK_AADT_PCT] * " + truckFactor + ")", "VB", "")

print "Calculating Cars Per Minute Per Lane...."

arcpy.CalculateField_management(TxDOT_Roadway_Linework_Routed,

"FORECAST_ALTERNATIVE_CARS_PER_MIN", "((([ADT_DESGN] + [FORECAST_ALTERNATIVE_TRUCKS]) * ( [K_FAC] *.01)) / ([NUM_LANES] * " + capacityFactor + ")) / 60", "VB", "")

print "Calculating Car Spacing...."

arcpy.CalculateField_management(TxDOT_Roadway_Linework_Routed, "FORECAST_ALTERNATIVE_CAR_SPACE", "((" + analysisUnit + ") - ((((([ADT_DESGN] +
```

```
[FORECAST_ALTERNATIVE_TRUCKS]) * ( [K_FAC] *.01)) / ([NUM_LANES] * " + capacityFactor + ")) / 60) *
[FORECAST_ALTERNATIVE_AVG_CAR_LENGTH])) / [FORECAST_ALTERNATIVE_CARS_PER_MIN]", "VB", "")

print "Script Complete"
```