**Constraint Analyzer Project**

**Data Collection:**

* **Location of all the files mentioned in this document:**
* **S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Data**

Needed data – Transmission outages data, Constraint data, Auction Contingency Data, Auction Mapping Document

\***For now this project is only developed for ERCOT region, once completed it can be developed for other regions as well.**

**\*This document contains information about all the file extensions except .py extension for which documentation is done in CodeFiles\_Documentation.docx file**

**Source of Historical Constraint Data:**

* Location:
* S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\Constraint Data
* Data is taken from the years 2014 to 2019.
* **Source - Yes Energy** is the vendor from where Historical Constraint data is pulled out.

**Source of Mapping Document:**

* Location:
* S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\Mapping Documents
* These mapping documents are taken from ERCOT.
* The mapping document can be taken from any year (preferably latest year’s data should be taken) as it contains the list of constraint-contingency that have happened over the period.

**Source of Transmission Outage Data:**

* Location:
* S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\Transmission Outages
* Data is taken from the years 2014 to 2019
* Source of data from all the years is **Yes Energy** – data pulled using the Get\_Historical\_Trans\_Outage\_v1.py
* Link to .py file - S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Code Files\Get\_Historical\_Trans\_Outage\_v1.py

**uniquePairList2014-2019.xlsx:**

* Location:
* S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\UniqueConstraintContingencyPair\uniquePairList2014-2019.xlsx
* This file is the output of ConstraintContingencyUniquePairCreation.py file.
* This file contains unique pairs of constraint-contingency for the years 2014-2019.
* Sheet Constraint-Contingency:
* Columns named –constraint, contingency, hrs, % Hrs, Total $, Avg $, max $, Constraint ID, Contingen. ID, Reported Name, Source Shift Factor, Sink Shift Factor, Spread Shift Factor are already present in the output obtained by python file.
* Column named – Code is created manually as follows:

1. Insert 3 columns next to ‘Constraint’ column in the excel sheet.
2. Select ‘Constraint’ column of the excel sheet.
3. Go to Data (Excel Ribbon) 🡪 Text to Column 🡪Select delimited 🡪 Click Next 🡪 Check space (uncheck all others) 🡪 Click Next 🡪 Click on Destination (in Convert Text to Columns Wizard – Step 3 of 3) 🡪 Select 4 newly inserted columns in the sheet after the ‘Constraint’ column 🡪 Click Finish
4. Pick the 3rd newly inserted column and name that column as ‘Code’. Delete the remaining columns.

* In a new sheet, copy this file “S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\Mapping Documents\2019.SEP.Monthly.Auction.MappingDocument.xlsx” and name it as “AuctionMapping2019SEP”
* In a new sheet, copy this file “S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\Mapping Documents\2019.SEP.Monthly.Auction.Contingencies.CSV” and name it as “AuctionContingency2019SEP”
* In this sheet do the following steps:

1. Select ‘DeviceName’ column of the excel sheet.
2. Go to Data (Excel Ribbon) 🡪 Text to Column 🡪Select delimited 🡪 Click Next 🡪 Check space (uncheck all others) 🡪 Click Next 🡪 Click on Destination (in Convert Text to Columns Wizard – Step 3 of 3) 🡪 Select 4 columns in the sheet after the ‘Action’ column 🡪 Click Finish
3. Name the first column as “From Bus Number”.
4. Name second column as “From Bus Name”.
5. Name the third column as “To Bus Number”.
6. Name the fourth column as “To Bus Name”.
7. Name the fifth column as “Circuit Id”.

* Save the excel file.
* This excel file is used as an input by ConstraintContingencyMapping.py

**uniquePairList2014-2019Changed.xlsx:**

* Location:
* S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\UniqueConstraintContingencyPair\uniquePairList2014-2019Changed.xlsx
* This file is the output of ConstraintContingencyMapping.py file.
* This file contains unique pairs of constraint-contingency for the years 2014-2019 in sheet ‘Constraint-Contingency’, auction mapping data of year 2019 September in ‘AuctionMapping2019SEP’ sheet and auction contingency data for the year 2019 September in sheet ‘AuctionContingency2019SEP’.
* Sheet Constraint-Contingency:
* Columns named –constraint, contingency, hrs, % Hrs, Total $, Avg $, max $, Constraint ID, Contingen. ID, Reported Name, Source Shift Factor, Sink Shift Factor, Spread Shift Factor are already present in the output obtained by python file.
* Column named – contingency\_from\_bus\_number, contingency\_from\_bus\_name, contingency\_to\_bus\_number, contingency\_to\_bus\_name, contingency\_circuit\_id, constraint\_from\_bus\_number, constraint\_from\_bus\_name, constraint\_to\_bus\_ number, constraint\_to\_bus\_name, constraint\_circuit\_id are fetched using VLOOKUP formula
* contingency\_from\_bus\_number:

=IFERROR(VLOOKUP(C3,AuctionContingency2019SEP!$A:$I,5,FALSE)," ")

* contingency\_from\_bus\_name:

=IFERROR(VLOOKUP(C3,AuctionContingency2019SEP!$A:$I,6,FALSE)," ")

* contingency\_to\_bus\_number:

=IFERROR(VLOOKUP(C3,AuctionContingency2019SEP!$A:$I,7,FALSE)," ")

* contingency\_to\_bus\_name:

=IFERROR(VLOOKUP(C3,AuctionContingency2019SEP!$A:$I,8,FALSE)," ")

* contingency\_circuit\_id:

=IFERROR(VLOOKUP(C3,AuctionContingency2019SEP!$A:$I,9,FALSE)," ")

* constraint\_from\_bus\_number:

=IFERROR(VLOOKUP(B2,AuctionMapping2019SEP!$C:$J,4,FALSE)," ")

* constraint\_from\_bus\_name:

=IFERROR(VLOOKUP(B2,AuctionMapping2019SEP!$C:$J,5,FALSE)," ")

* constraint\_to\_bus\_ number:

=IFERROR(VLOOKUP(B2,AuctionMapping2019SEP!$C:$J,6,FALSE)," ")

* constraint\_to\_bus\_name:

=IFERROR(VLOOKUP(B2,AuctionMapping2019SEP!$C:$J,7,FALSE)," ")

* constraint\_circuit\_id:

=IFERROR(VLOOKUP(B2,AuctionMapping2019SEP!$C:$J,8,FALSE)," ")

* Columns named

1. Interface Name is fetched using formula –

=IF(AND(O2<>" ",P2<>" ",Q2<>" ",R2<>" ",T2<>"",U2<>" ",V2<>" ",W2<>" "),CONCATENATE(U2,"-",W2," ","FLO"," ",P2,"-",R2),IF(AND(T2<>" ",U2<>" ",V2<>" ",W2<>" "),CONCATENATE(U2,"-",W2," ","FLO"," ","BASE"),"NaN"))

1. Element a (Contingency) is fetched using formula-

=IF(AND(O2<>" ",P2<>" ",Q2<>" ",R2<>" "),CONCATENATE("BRANCHOPEN"," ",O2," ",Q2," ",S2),"NaN")

1. Element b (Monitored) is fetched using formula –

=IF(AND(T2<>" ",U2<>" ",V2<>" ",W2<>" "),CONCATENATE("BRANCH"," ",T2," ",V2," ",X2),"NaN")

1. Meter Far is given a value – “NO”
2. Weight is given a value – “1”

* Copy the formula to all the cells
* Save the excel file.
* This file is used as an input by the ConstraintContingencySeparation.py

**uniquePairList2014-2019Separation.xlsx:**

* Location:
* S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\UniqueConstraintContingencyPair\uniquePairList2014-2019Separation.xlsx
* This file is the output of ConstraintContingencySeparation.py file.
* Sheet ‘mappedConstraints’ – all the columns are already present in the output obtained by python file.
* Sheet ‘mappedContingency’ - all the columns are already present in the output obtained by python file.
* Sheet ‘unmappedConstraints’ - all the columns are already present in the output obtained by python file except ‘match’ which is created as follows:
* Select ‘Constraint’ column of the excel sheet.
* Go to Data (Excel Ribbon) 🡪 Text to Column 🡪Select delimited 🡪 Click Next 🡪 Check space (uncheck all others) 🡪 Click Next 🡪 Click on Destination (in Convert Text to Columns Wizard – Step 3 of 3) 🡪 Select 3 newly columns in the sheet after the last column 🡪 Click Finish
* Pick the 3rd newly inserted column and name that column as ‘match’. Delete the remaining columns.
* Sheet ‘unmappedContingency’ - all the columns are already present in the output obtained by python file.
* Sheet ‘mappedBoth’ - all the columns are already present in the output obtained by python file.
* Save the excel file.
* This file is used as an input by the UnmappedConstraintsApproximateStringMatching.py file.

**UnmappedConstraintsApproximateStringMatching.xlsx:**

* Location:
* S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\UniqueConstraintContingencyPair\uniquePairList2014-2019Separation.xlsx
* This file is the output of UnmappedConstraintsApproximateStringMatching.py file.
* Manual processing carried out in this file is as follows:
* Create a new sheet named ‘manual\_mapped’
* Copy ‘constraint’, ‘match’, ‘matching\_code’ to the new sheet.
* Manually check strings separated by ‘\*’ in the matching code column with the match column for similarity.
* If they mostly match, check for the from and to bus name in the auction mapping document of the year 2019 September and compare it with first part of the ‘constraint’ column.
* If all the checks are correct or mostly match keep the row and delete the rows that don’t match.
* In the matching code column, keep only that string which matches the most removing the ‘\*’ as well.
* In the sheet named “All” create a column named ‘matching\_code\_mapped’ which is filled using this formula: =IFERROR(VLOOKUP(D2,manual\_mapped!$A:$C,3,FALSE)," ")
* Sheet named ‘unmapped’ consist the constraints that remained unmapped after manual mapping. Filter was applied in the ‘All’ sheet and then the data obtained was copy pasted into ‘unmapped’ sheet.
* Sheet named ‘total\_$\_analysis’ contains pivot table to analyze in what total\_$ range does the unmapped constraints mostly fall.
* Sheet ‘AuctionMappingSEP2019’ is copied from “S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\Mapping Documents\2019.SEP.Monthly.Auction.MappingDocument.xlsx”
* This file is used as one of the inputs in the ConstraintContingencyFinal.py file.

**ConstraintContingencyFinal.xlsx:**

* Location:
* S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Constraint-Project\Data\UniqueConstraintContingencyPair\Constraint-ContingencyFinal.xlsx
* This file is the output of ConstraintContingencyFinal.py file.
* Sheet ‘mappedConstraints’ contains final list of mapped constraints
* Sheet ‘unmappedConstraints’ contains final list of unmapped constraints
* Sheet ‘mappedContingency’ contains final list of mapped contingencies
* Sheet ‘unmappedContingency’ contains final list of unmapped contingencies
* Sheet ‘mappedBoth’ contains final list of mapped Constraint and Contingency pair.

**Source of PowerWorldFormat.xlsx:**

* Location - S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Data\PowerWorldFormat.xlsx
* This file contains data only for the year 2018.
* Sheet name – Mapped
* This sheet contains the list of all the mapped contingency-monitored element pair i.e. for an interface name to be included either a monitored line from bus number was present or monitored line from bus number and contingency from bus number was present.
* All the columns of this sheet are created using code file named ConstraintContingencyMapping.py
* Sheet name – Unmapped
* This sheet contains list of all the unmapped contingency-monitored element pair i.e. a pair is included if a monitored line from bus number is not present and a contingency from bus number is present or a monitored line from bus number and contingency from bus number both are not present.
* All the columns in this sheet are created using a code file named ConstraintContingencyMapping.py

**Source of PowerWorldFormat2015-2018.xlsx:**

* Location - S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Data\PowerWorldFormat.xlsx
* This file contains data only for the years 2015-2018.
* Sheet name – Mapped
* This sheet contains the list of all the mapped contingency-monitored element pair i.e. for an interface name to be included either a monitored line from bus number was present or monitored line from bus number and contingency from bus number was present.
* All the columns of this sheet are created using code file named ConstraintContingencyMapping.py
* Sheet name – Unmapped
* This sheet contains list of all the unmapped contingency-monitored element pair i.e. a pair is included if a monitored line from bus number is not present and a contingency from bus number is present or a monitored line from bus number and contingency from bus number both are not present.

All the columns in this sheet are created using a code file named ConstraintContingencyMapping.py (second run).

**Preprocessing of Transmission Outages Files:**

Preprocessing of the files listed below must be done before using it as an input. All files are:

1. S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Data\Outages + Constraint\ERCOT\_TransmissionOutage\_2014-01-01\_to\_2014-12-31.xlsx
2. S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Data\Outages + Constraint\ERCOT\_TransmissionOutage\_2015-01-01\_to\_2015-12-31.xlsx
3. S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Data\Outages + Constraint\ERCOT\_TransmissionOutage\_2016-01-01\_to\_2016-12-31.xlsx
4. S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Data\Outages + Constraint\ERCOT\_TransmissionOutage\_2017-01-01\_to\_2017-12-31.xlsx
5. S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Data\Outages + Constraint\ERCOT\_TransmissionOutage\_2018-01-01\_to\_2018-12-31.xlsx
6. S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Data\Outages + Constraint\ERCOT\_TransmissionOutage\_2019-01-01\_to\_2019-07-24.xlsx

Following step are to be performed on all the files mentioned above:

1. Select ‘reported name’ column of the excel file.
2. Go to Data (Excel Ribbon) 🡪 Text to Column 🡪Select delimited 🡪 Click Next 🡪 Check Space (uncheck all others) 🡪 Click Next 🡪 Click on Destination (in Convert Text to Columns Wizard – Step 3 of 3) 🡪 Select 4 columns in the sheet after the last column i.e. ‘RELIABILITY\_SCORE’ 🡪 Click Finish
3. In the newly created columns, only keep column 1st and 3rd. Delete the other two columns.
4. Name the newly created 1st column as ‘from’ and 3rd column as ‘to’.
5. Also, copy this excel file - S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Data\Mapping Documents\2019.JUL.Monthly.Auction.MappingDocument.xlsx, sheet name – Lines and paste it to a new sheet of transmission outage file. (this must be done for all the transmission files)

**Source of all UniqueTransmissionOutagesMappedYEAR.xlsx (where YEAR can be – 2014,2015,2016,2017,2018,2019)**

* Location: S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Data
* These files are created using a python file named – TransmissionOutagesMapping.py
* Rename sheet 1 as ‘Outages’
* Rename sheet 2 as AuctionMapping2019JUL\_LINES
* Copy - S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Data\Mapping Documents\2019.JUL.Monthly.Auction.MappingDocument.xlsx, sheet name = Autos and paste it into sheet 3 of this file, rename it as ‘AuctionMapping2019JUL\_AUTOS
* Delete the index column in ‘Outages’ sheet and ‘AuctionMapping2019JUL\_LINES sheet created by the python code.
* In the ‘Outages’ sheet Columns named – iso, facility, fromstation, tostation, kv, fromzone, tozone, facility\_type, type, type\_detail, status, status\_detail, startdate, enddate, planned\_startdate, planned\_enddate, open\_close, ticketid, facilityid, lastchangedate, publishdate, reported\_name, fromstationid, tostationid, reliability\_score, reliability\_accuracy, from, to are all fetched by the python code file itself.
* In the AuctionMapping2019JUL\_AUTOS sheet, insert new columns as follows:

1. Select ‘Operations\_Name’ column of the excel sheet.
2. Go to Data (Excel Ribbon) 🡪 Text to Column 🡪Select delimited 🡪 Click Next 🡪 Check Other – enter underscore symbol (uncheck all others) 🡪 Click Next 🡪 Click on Destination (in Convert Text to Columns Wizard – Step 3 of 3) 🡪 Select 2 columns in the sheet after the last column i.e. ‘ID’ 🡪 Click Finish
3. Name the 1st column as ‘Operations’ and delete the rest.
4. Insert a column next to ‘Operations\_Name’ column and move the newly created column ‘Operations’ next to it.
5. Insert a column next to ‘Operations’ named ‘Combine’ and use this formula =C2&"|"&E2

To obtain the values for this column.

* In the ‘Outages’ sheet Columns which are created manually after executing python code to obtain this file are:

1. combine: formula used for obtaining column value

=AA2&"|"&AB2

1. from\_bus\_number: formula used for obtaining column value =IF(H2="XFMR",IFERROR(VLOOKUP(AC2,AuctionMapping2019JUL\_AUTOS!$D:$J,3,FALSE)," "),IFERROR(VLOOKUP(AB2,AuctionMapping2019JUL\_LINES!$C:$J,4,FALSE)," "))
2. from\_bus\_name: formula used for obtaining column value =IF(H2="XFMR",IFERROR(VLOOKUP(AC2,AuctionMapping2019JUL\_AUTOS!$D:$J,4,FALSE)," "),IFERROR(VLOOKUP(AB2,AuctionMapping2019JUL\_LINES!$C:$J,5,FALSE)," "))
3. to\_bus\_number: formula used for obtaining column value =IF(H2="XFMR",IFERROR(VLOOKUP(AC2,AuctionMapping2019JUL\_AUTOS!$D:$J,5,FALSE)," "),IFERROR(VLOOKUP(AB2,AuctionMapping2019JUL\_LINES!$C:$J,6,FALSE)," "))
4. to\_bus\_name: formula used for obtaining column value =IF(H2="XFMR",IFERROR(VLOOKUP(AC2,AuctionMapping2019JUL\_AUTOS!$D:$J,6,FALSE)," "),IFERROR(VLOOKUP(AB2,AuctionMapping2019JUL\_LINES!$C:$J,7,FALSE)," "))
5. circuit\_id: formula used for obtaining column value =IF(H2="XFMR",IFERROR(VLOOKUP(AC2,AuctionMapping2019JUL\_AUTOS!$D:$J,7,FALSE)," "),IFERROR(VLOOKUP(AB2,AuctionMapping2019JUL\_LINES!$C:$J,8,FALSE)," "))
6. reported\_name\_duplicate: copy and paste column values of ‘reported\_name’ column.

* In the ‘AuctionMapping2019JUL\_LINES’ sheet create a new column next to the last column named ‘circuit\_id’ and name this new column as ‘outages\_reported\_names\_mapped’. The values for this column are obtained using the following formula:

=IFERROR(VLOOKUP(F2,Outages!$AD:$AI,6,FALSE)," ")

* In the ‘AuctionMapping2019JUL\_AUTOS’ sheet create a new column next to the last column named ‘circuit\_id’ and name this new column as ‘outages\_reported\_names\_mapped’. The values for this column are obtained using the following formula:

=IFERROR(VLOOKUP(F2,Outages!$AD:$AI,6,FALSE)," ")

**Source of UniqueTransmissionOutagesMapped2014-2019.xlsx:**

* Location: S:\asset ops\GO\_Group\Interns\2019\Anubha\Constraint Project\Data\UniqueTransmissionOutagesMapped2014-2019.xlsx
* This file is an output file which is generated by a python code file named – UniqueTransmissionOutagesConcat.py

**Why is mapping done?**

The data that we get from Yes Energy or Energy Velocity consists the names of Monitored Element and Contingency whereas PowerWorld only consists of numbers i.e. From Bus Number and To Bus Number. So, mapping of names to numbers needs to be done to perform calculations of sensitivity factors such as PTDF, LODF and TLR.

Similar process must be carried out for transmission outages to determine which transmission line is open/close in the power system network.

**Parameters that are thought to be considered in training dataset:**

* All transmission outages as different parameter (can have a 0/1 value or LODF)
* Load
* Fuel Price
* PTDF
* On Peak/Off Peak
* Generators
* Date
* Month
* Year
* Week
* Time
* Shadow Price