

Treasure Coast Regional Planning Model 4.0

Enhanced Program Documentation

Prepared for:

Florida Department of Transportation District 4
3400 West Commercial Boulevard
Fort Lauderdale, FL 33309



Prepared by:

The Corradino Group, Inc.
200 S. 5th Street, Suite 503N
Louisville, KY 40202



July 2017

Treasure Coast Regional Planning Model, Version 4 Summary Program Documentation

Introduction

This report provides summary descriptions of the most important elements of the Treasure Coast Regional Planning Model, Version 4 (TCRPM4). The report provides an overview description of the modeling process. It does not replace the two primary model documents, *Treasure Coast Regional Planning Model 4.0*, *Activity-Based Model Users Guide*, and *Treasure Coast Regional Planning Model -- TCRPM 4.0, Coordinated Travel – Regional Activity Based Modeling Platform (CT-RAMP), Model Development Report*. The reader should refer to these reports for more information on the details of the model. Furthermore, readers with access to the Cube Voyager software and the TCRPM4 application probably will want to “drill-down” in the software to discover even more model implementation details. Those without access to the software or who would prefer not to wade through the details should find this report to be useful.

This report does not attempt to educate the reader on the basics of travel demand modeling, and presupposes that the reader has a basic understanding of how travel models work. Similarly, this report presupposes that the reader is familiar with flowcharts and computer scripting.

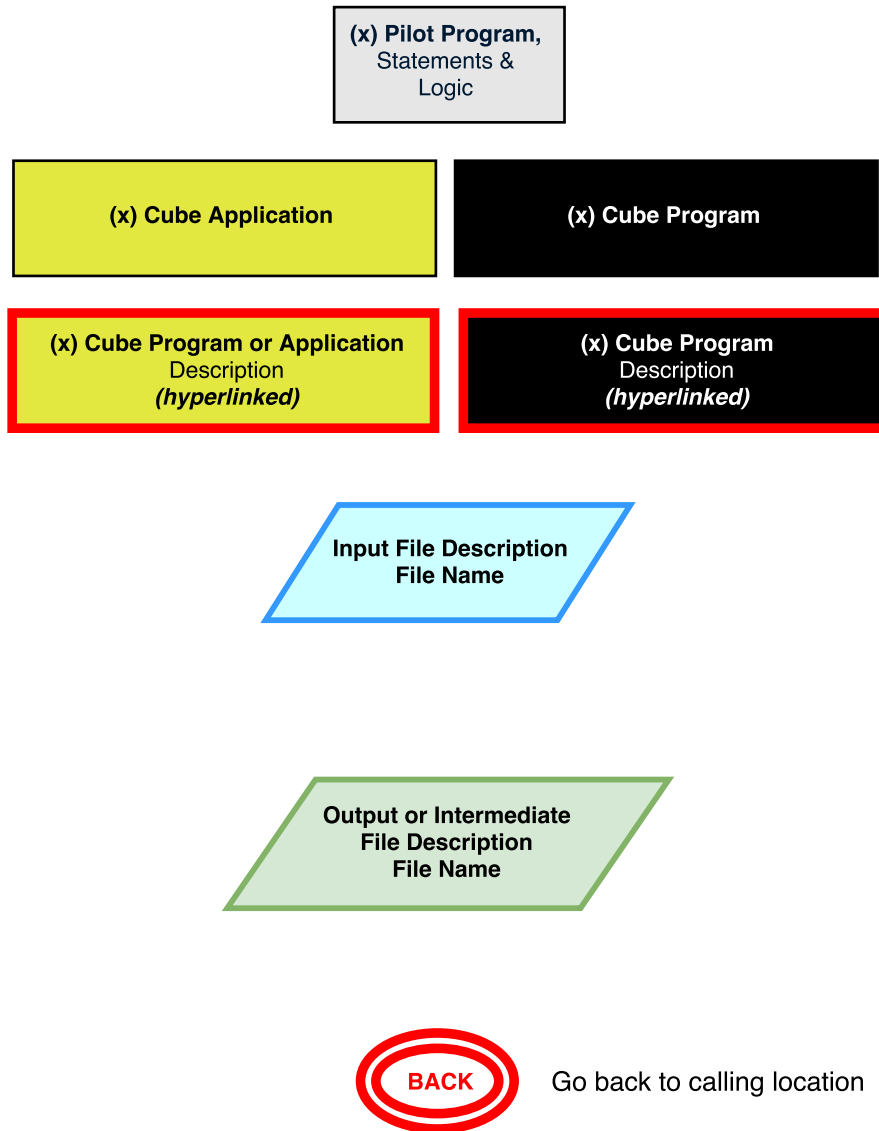
Using This Report

This report can be printed and used as an ordinary reference report. But, it is best used as a hyperlinked document, viewed on-line or as “pdf” file. Viewed through Adobe Reader, the document contains bookmarks, which will allow the reader to jump to any section of the report. The main organization of the report is that of a flowchart. The “Outline – Top Level Flowchart” is the most convenient tool for navigating the report. Besides providing a convenient way of navigating the report, it parallels the Cube Voyager top-level flowchart, presenting the main model steps. Notice that the flowcharts and programs use the same numbering system as the Cube Voyager software program flowchart. For example, step 4 of the top-level flowchart is “Network”, and the details appear in (4) NETWORK APPLICATION. Hyperlinks allow the reader to examine details of the most important sub-applications. In future versions of this report, more hyperlinked descriptions may be developed.

The next section in the report, following this introduction, is the flowchart key, which describes the meaning of the colors and shapes used in the flowchart.

Following the key is the Top Level Flowchart. In the TCRPM 4 Top Level Flowchart, every major process is hyperlinked. Clicking on the boxes outlined in red will take you to a more detailed flowchart (sub-application). Some of the hyperlinks in sub-applications will take you to a text-based description of a process, which is simplified Cube code. Presenting these descriptions as flowcharts would have required many pages, and it was the consultant’s judgment that the pseudo code would be easier to understand. Note also that every flowchart and text description contains a “BACK” symbol, which returns to the previous hyperlink location.

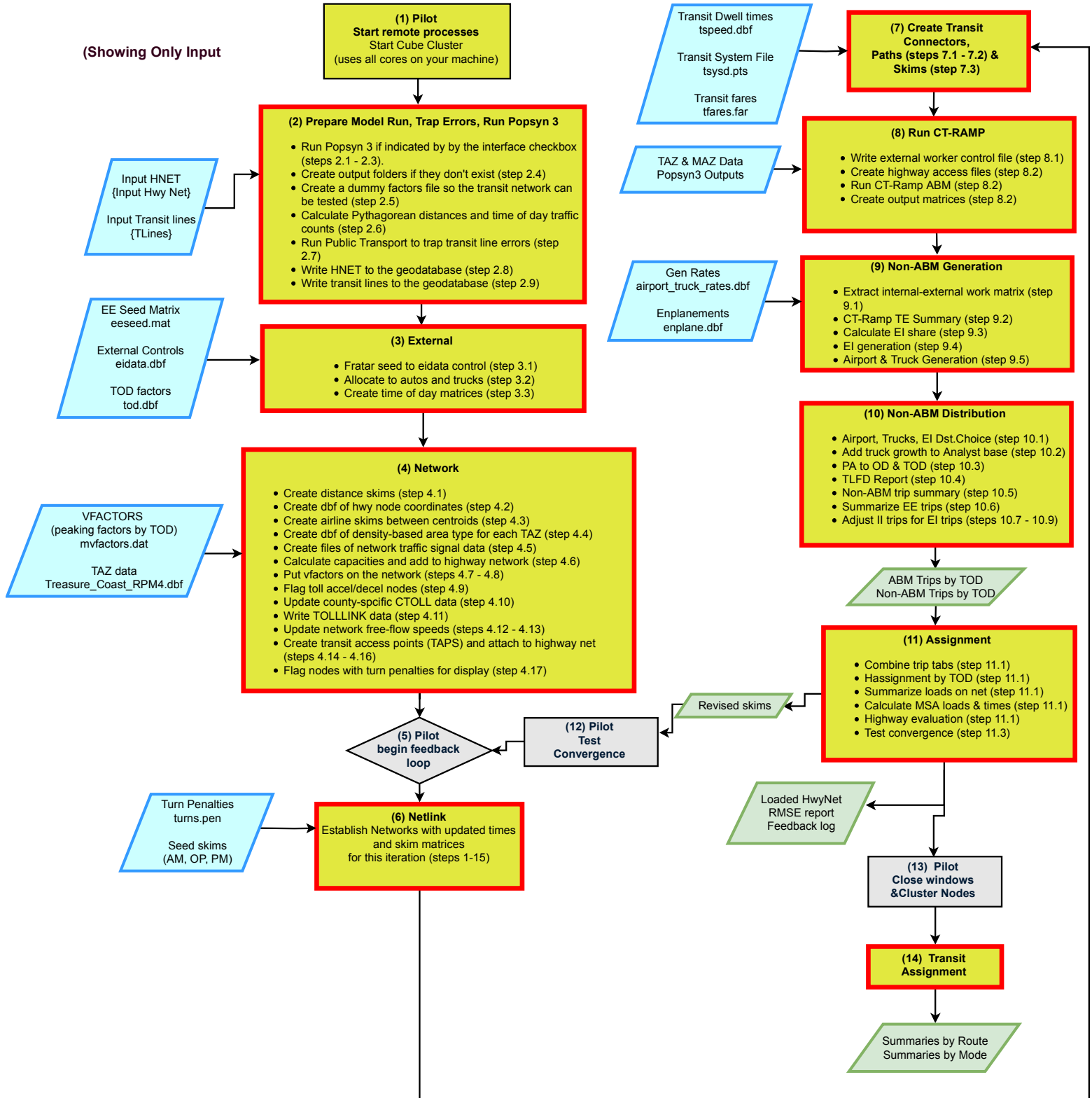
FLOWCHART KEY



(x) = Cube Flowchart Sequence number

TCRPM 4 Top Level Flowchart

(Showing Only Input)



(1) Start Remote Processes



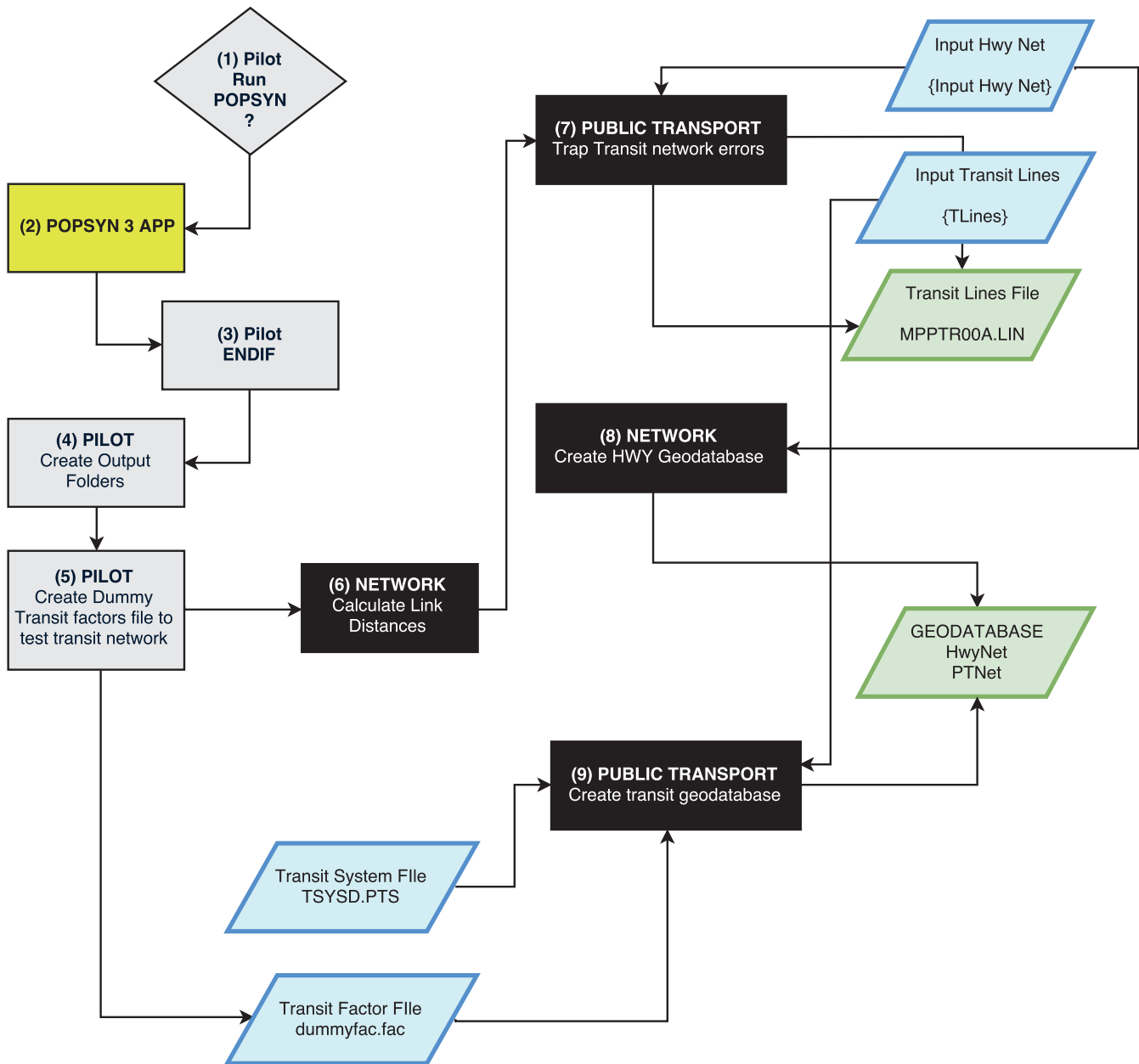
(1) PILOT

- Close all Cluster Nodes
- Start & Hide all Cluster Nodes

Use all available cluster nodes (cores)

This application calls the Cube Cluster program to close any Cluster nodes that may be active, and then starts new Cluster nodes. A Cluster node is a microprocessor thread, which enables parallel processing.

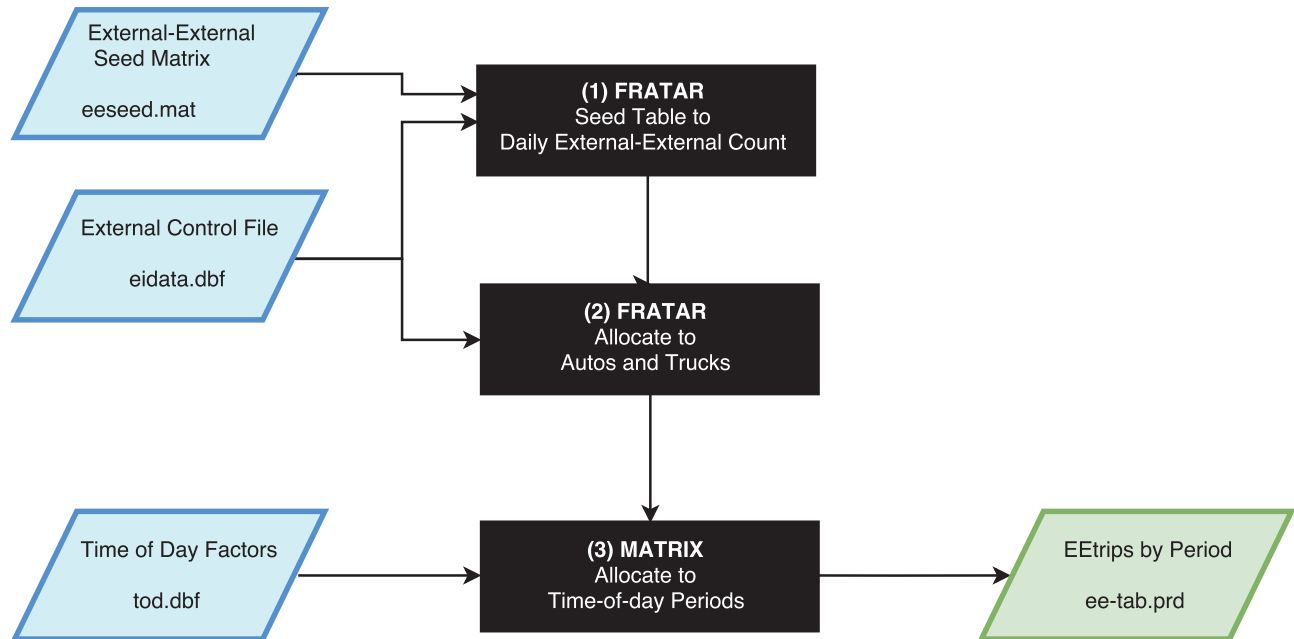
(2) Prepare Model Run, Trap Errors, Run POPSYN3



This application does the following:

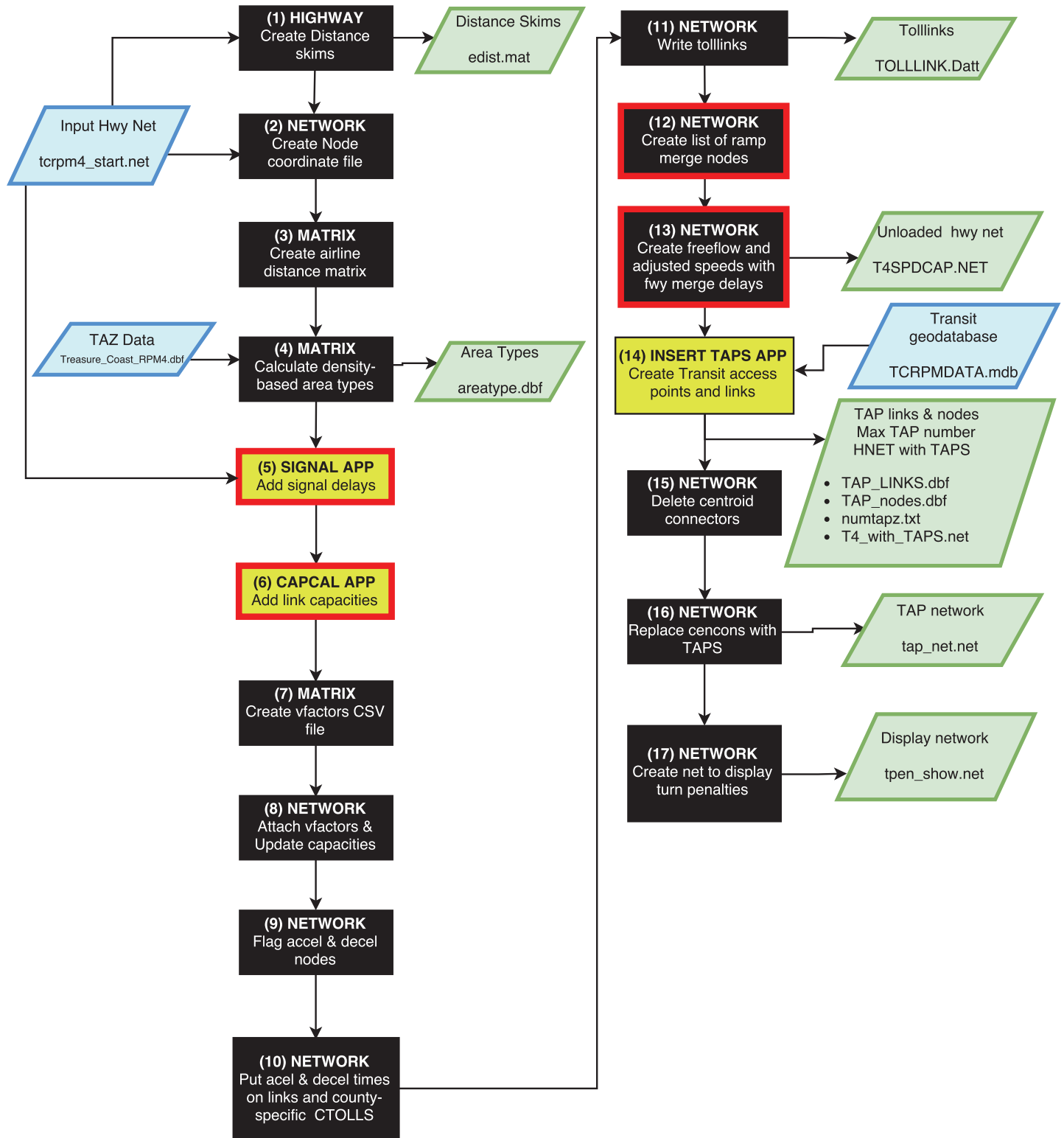
- . Runs the Population synthesizer if the Cube Key {popsyn} is true (interface checkbox checked)
- . Creates the output folder named in the Cube Key {OUTDIR}
- . Deletes unneeded output files which may have been created in earlier model runs
- . Creates the feedback.log file, which reports progress for the model feedback loop on travel time
- . Creates the runtime.prn file which records the timing of major model steps
- . Creates the Title.Mas file and inserts the value of the Cube Key {DESCR}, which is used in reporting.

(3) External Model (EE Trip Table)



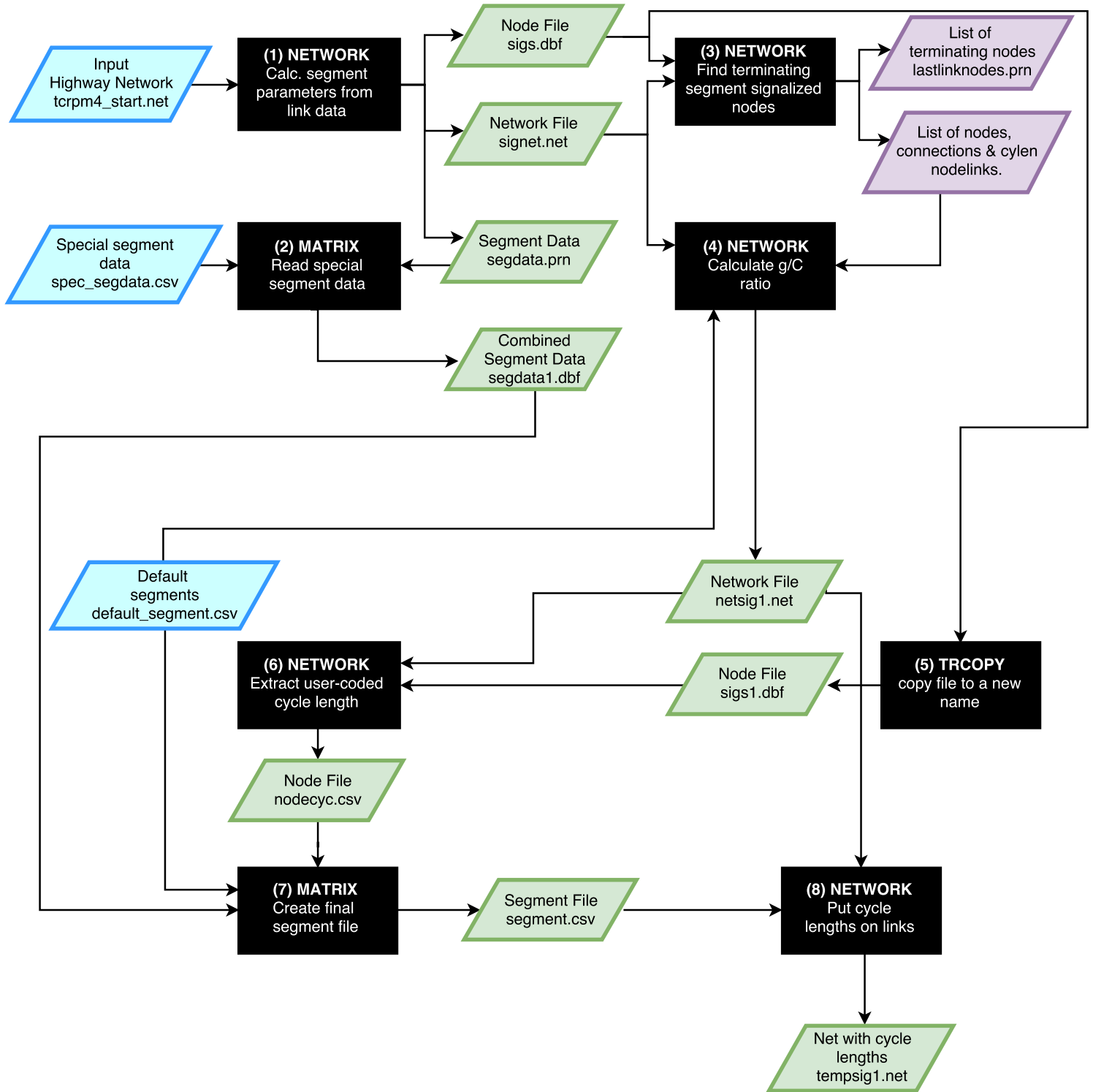
This application begins with three input files: the external-external trip table seed matrix, which was estimated from the best available survey data, the external control file that specified "target" external station auto and truck percentages, and the time-of-day (TOD) factors file, which specifies the percentage of trips at the external stations (and other trip purposes) for the five time periods. The first Fratar program uses proportional fitting to adjust the seed trip table to match external volumes for the target year. The second Fratar program uses proportional fitting to match auto and truck volumes at the stations. The Matrix program allocates the trips to the five modeling time periods.

(4) NETWORK APPLICATION



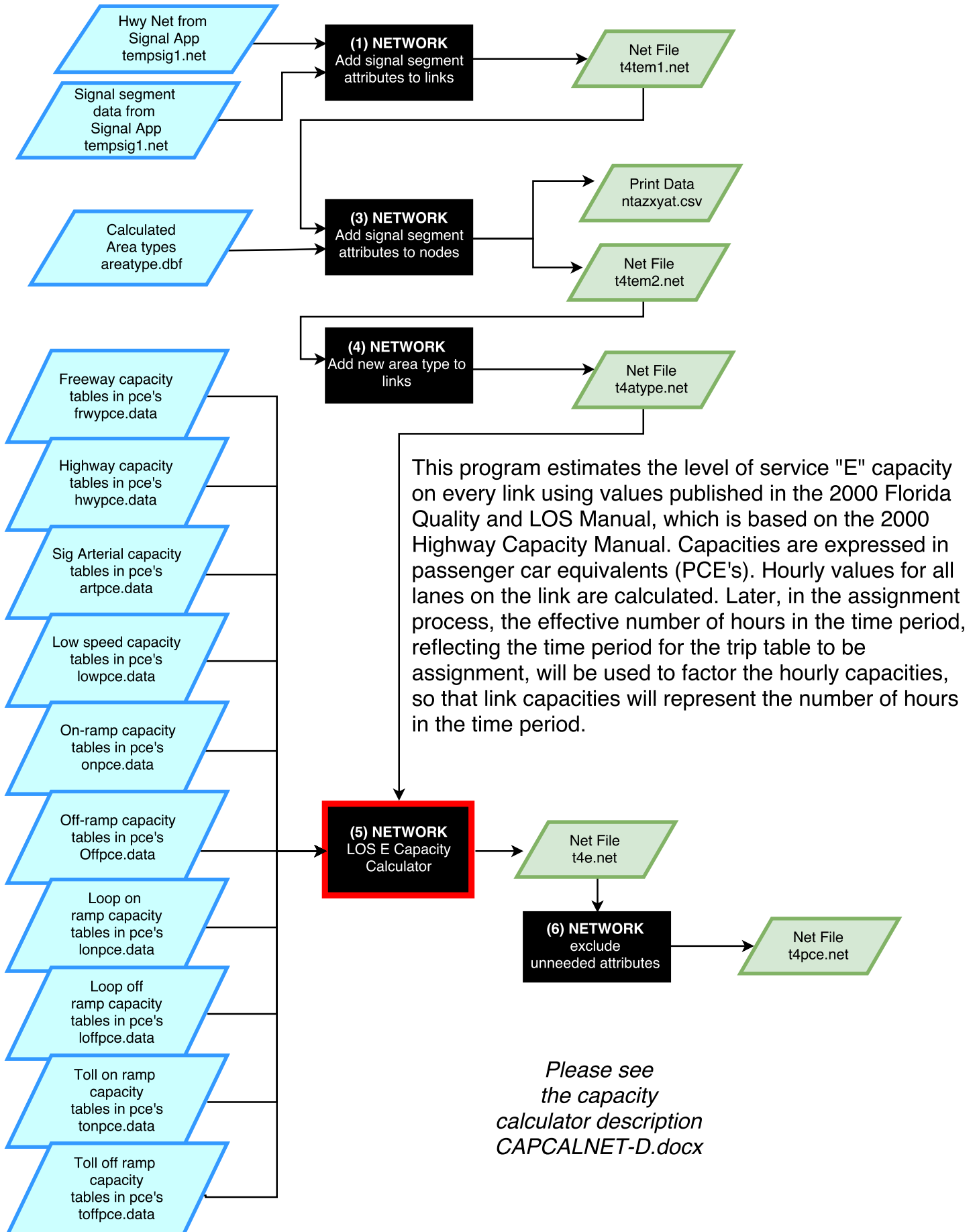
This application performs all of the processing needed to create the highway, and add transit access points (TAP's), which form the basis for the transit networks and trip tables. Included in this application is the calculation of highway link speeds, travel times, intersection delays, and highway link capacities by time of day.

(4.5) SIGNAL APPLICATION (evaluate signal parameters)



This application evaluates all data provided to describe travel control devices and adds the basis unloaded delays to the network. There is a hierarchy used in these calculations. If intersection-specific parameters are provided, they are used. Otherwise, if signal segment data are provided (a group of coordinated signals) they are used. Otherwise, global system defaults are used. The presence of signals are indicated on node records with a non-zero signal location code (SIGLOC). Links that are part of a coordinated signal system have a non-zero segment identifier (SEGID). Signal delays are a function of cycle length, green to cycle ratio (G/C), and degree of coordination.

(4.6) CAPCAL APPLICATION (calculate capacities)



(4.6.5) CAPCALNET-D.S 'LOS E Capacity Calculator'

VOYAGER NETWORK PROGRAM

Calculates hourly capacities for all links in the network

Input Files

- LINKI=T4ATYPE.NET (highway network)
- LOOKUPI = capacity tables
 1. FRWYPCE.DAT – freeway capacities
 2. HWYPCE.DAT – uninterrupted flow highway capacities
 3. ARTPCE.DAT – signalized arterial capacities
 4. LOWPCE.DAT – low speed roadway capacities
 5. ONPCE.DAT – on-ramp capacities
 6. OFFPCE.DAT- off ramp capacities
 7. LONPCE.DAT – loop on-ramp capacities
 8. LOFFPCE.DAT – loop off-ramp capacities
 9. TONPCE.DAT – toll on-ramp capacities
 10. TOFFPCE.DAT – toll off-ramp capacities

Output Files

- NETO=T4E.NET (Loaded Network with capacities)
 - Exclude attribute CAPALL

Process Each Link (LINKMERGE) the following logic is applied to each link:

- num_lanes=lanes (directional)
- if the link is signalized (SEGID>0 & NUMSIG>0)
 - SIG_RATE=NUMSIG/LENGTH (signals per mile for each segment)
 - SIG_SPACE=1/SIG_RATE (space between signals for each segment)
- Define Several Capacity Calculator Attributes to be used later
- HOV Facilities (FTC2=81-86)
 - HOV=1
 - FTC1=80
- Toll Facilities (FTC2=91-95)
 - KTOLL=1
 - FTC1=90
 - Toll Plazas (FTC2=95)
 - TOLLPLAZA=1
- Freeway links of all types (FTC2=11,12,81-82,91-92,95)
 - FRWY=1
 - Interstates (FTC2=11,12)
 - FTC1=10
- CDs, Expressways and Parkways (FTC2=21)
 - UNINTRP=1
 - FTC1=20
- High Speed Arterials (FTC2=41)

- FTC1=40
- Low Speed Collectors (<35 mph) (FTC2=61)
 - FTC1=60
 - LOWSPD=1
- All Ramps (FTC2=71-75,83-86,93-94)
 - RAMPS=1
 - All Non-HOV and Non-Toll Ramps (FTC2=71-75)
 - FTC1=70
 - Non-Loop On-Ramps, Excluding Freeway-To-Freeway (FTC2=71)
 - ON=1
 - Loop On-Ramps (FTC2=72)
 - ONLOOP=1
 - Off-Ramps, Excluding Freeway-To-Freeway (FTC2=73)
 - OFF=1
 - Loop Off-Ramps (FTC2=74)
 - OFFLOOP=1
 - Freeway-To-Freeway Ramps (FTC2=75)
 - FRWY2FRWY=1
 - FRWY=1 (treat as fwy segments for capacity calculation purposes)
 - Peak Only HOV Ramps (FTC2=83-85)
 - HOVPEAK=1
 - All-Day HOV Ramps (FTC2=86)
 - HOVDAY=1
 - Toll Ramps (FTC2=93,94)
 - Toll On-Ramps (FTC2=93)
 - TOLLON=1
 - Toll Off-Ramps (FTC2=94)
 - TOLLOFF=1
- IMPORTANT NOTE: (1) DIVIDED, (2) LEFTTURN and (3) TWOWAY should be directly coded
- Centroid Connectors (51=internal, 52=external) (FTC2=51-52)
 - FTC1=50
- Transit-Only Links (FTC2=49) (for local bus routes)
 - FTC1=49
- Transit Only Micro Connectors (FTC2=59)
 - FTC1=59
- Fixed-Guideway Links (FTC2=69)
 - FTC1=69
- Links that are not centroid connectors and not transit-only (!(FTC2=51,52,49,59,69))
 - Signalized links (SEGID>0 & NUMSIG>0)
 - $SIG_RATE = NUMSIG / LENGTH$ (signals per mile for each segment)
 - $SIG_SPACE = 1 / SIG_RATE$ (space between signals for each segment)
 - UNINTERRUPTED CHECK – Keys: {UNINTSPEED}, {SIG-SPACE}
 - Interrupted and signalized (NUMSIG>0 & FRWY=0 & RAMPS=0 & POSTED_SPD>={UNINTSPEED} & SIG_SPACE>{SIG-SPACE})

- UNINTRP=1
 - FTC1=20
 - FTC2=21 (note: overrides input coded value)
- Uninterrupted and unsignalized (NUMSIG=0 & FRWY=0 & RAMPS=0 & POSTED_SPD>={UNINTSPEED})
 - UNINTRP=1
 - FTC1=20
 - FTC2=21 (note: overrides input coded value)
- LOW SPEED CHECK
- Low-speed links (UNINTRP=0 & RAMPS=0 & POSTED_SPD<35 & KTOLL=0 & HOV=0 & !(FTC2=49,59,69))) LOWSPD=1
 - FTC1=60
 - FTC2=61 (note: overrides input coded value)
- Define Lookup Tables
- Hourly Freeway Capacities FRWYPCE(urb/rur, lanes), not interpolated
 LOOKUP, NAME=FRWYPCE, LOOKUPI=1, ;(file 1)
 LOOKUP[1]=1, RESULT=2, ; curve 1 lookup from col 1 return value from col 2 (urban)
 LOOKUP[2]=1, RESULT=3, ; curve 2 lookup from col 1 return value from col 3 (rural)
 FAIL=0,0,0,
 INTERPOLATE=N
- Hourly Uninterrupted Highway Capacities HWYPCE (1, lanes), not interpolated
 LOOKUP, NAME= HWYPCE, LOOKUPI=2, ;(file 2)
 LOOKUP[1]=1, RESULT=2, ; curve 1 lookup from col 1 return value from col 2
 FAIL=0,0,0,
 INTERPOLATE=N
- Hourly Uninterrupted Arterial Capacities ARTPCE(areatype/signal rate, lanes), not interpolated
 LOOKUP, NAME= ARTPCE, LOOKUPI=3, ;(file 3)
 LOOKUP[1]=1, RESULT=2, ; curve 1 lookup from col 1 return value from col 2
 LOOKUP[2]=1, RESULT=3, ; curve 2 lookup from col 1 return value from col 3
 LOOKUP[3]=1, RESULT=4, ; curve 2 lookup from col 1 return value from col 4
 LOOKUP[4]=1, RESULT=5, ; curve 2 lookup from col 1 return value from col 5
 LOOKUP[5]=1, RESULT=6, ; curve 2 lookup from col 1 return value from col 6
 LOOKUP[6]=1, RESULT=7, ; curve 2 lookup from col 1 return value from col 7
 FAIL=0,0,0,
 INTERPOLATE=N
- Hourly Low Speed Capacities LOWPCE(NEWAREA, lanes), not interpolated
 LOOKUP, NAME= LOWPCE, LOOKUPI=4, ;(file 4)
 LOOKUP[1]=1, RESULT=2, ; curve 1 lookup from col 1 return value from col 2
 LOOKUP[2]=1, RESULT=3, ; curve 2 lookup from col 1 return value from col 3
 LOOKUP[3]=1, RESULT=4, ; curve 2 lookup from col 1 return value from col 4
 LOOKUP[4]=1, RESULT=5, ; curve 2 lookup from col 1 return value from col 5
 FAIL=0,0,0,
 INTERPOLATE=N
- Hourly On Ramp Capacities ONPCE(NEWAREA, lanes), not interpolated
 LOOKUP, NAME= ONPCE, LOOKUPI=5, ;(file 5)

LOOKUP[1]=1, RESULT=2, ; curve 1 lookup from col 1 return value from col 2
 LOOKUP[2]=1, RESULT=3, ; curve 2 lookup from col 1 return value from col 3
 LOOKUP[3]=1, RESULT=4, ; curve 2 lookup from col 1 return value from col 4
 LOOKUP[4]=1, RESULT=5, ; curve 2 lookup from col 1 return value from col 5
 LOOKUP[5]=1, RESULT=6, ; curve 2 lookup from col 1 return value from col 6
 FAIL=0,0,0,
 INTERPOLATE=N

- Hourly Off Ramp Capacities OFFPCE(NEWAREA, lanes), not interpolated
 LOOKUP, NAME= OFFPCE, LOOKUPI=6, ;(file 6)
 LOOKUP[1]=1, RESULT=2, ; curve 1 lookup from col 1 return value from col 2
 LOOKUP[2]=1, RESULT=3, ; curve 2 lookup from col 1 return value from col 3
 LOOKUP[3]=1, RESULT=4, ; curve 2 lookup from col 1 return value from col 4
 LOOKUP[4]=1, RESULT=5, ; curve 2 lookup from col 1 return value from col 5
 LOOKUP[5]=1, RESULT=6, ; curve 2 lookup from col 1 return value from col 6
 FAIL=0,0,0,
 INTERPOLATE=N
- Hourly On Loop Ramp Capacities LONPCE(NEWAREA, lanes), not interpolated
 LOOKUP, NAME= LONPCE, LOOKUPI=7, ;(file 7)
 LOOKUP[1]=1, RESULT=2, ; curve 1 lookup from col 1 return value from col 2
 LOOKUP[2]=1, RESULT=3, ; curve 2 lookup from col 1 return value from col 3
 LOOKUP[3]=1, RESULT=4, ; curve 2 lookup from col 1 return value from col 4
 LOOKUP[4]=1, RESULT=5, ; curve 2 lookup from col 1 return value from col 5
 LOOKUP[5]=1, RESULT=6, ; curve 2 lookup from col 1 return value from col 6
 FAIL=0,0,0,
 INTERPOLATE=N
- Hourly Off Loop Ramp Capacities LOFFPCE(NEWAREA, lanes), not interpolated
 LOOKUP, NAME= LOFFPCE, LOOKUPI=8, ;(file 8)
 LOOKUP[1]=1, RESULT=2, ; curve 1 lookup from col 1 return value from col 2
 LOOKUP[2]=1, RESULT=3, ; curve 2 lookup from col 1 return value from col 3
 LOOKUP[3]=1, RESULT=4, ; curve 2 lookup from col 1 return value from col 4
 LOOKUP[4]=1, RESULT=5, ; curve 2 lookup from col 1 return value from col 5
 LOOKUP[5]=1, RESULT=6, ; curve 2 lookup from col 1 return value from col 6
 FAIL=0,0,0,
 INTERPOLATE=N
- Hourly On Toll Ramp Capacities TONPCE(NEWAREA, lanes), not interpolated
 LOOKUP, NAME= TONPCE, LOOKUPI=9, ;(file 9)
 LOOKUP[1]=1, RESULT=2, ; curve 1 lookup from col 1 return value from col 2
 LOOKUP[2]=1, RESULT=3, ; curve 2 lookup from col 1 return value from col 3
 LOOKUP[3]=1, RESULT=4, ; curve 2 lookup from col 1 return value from col 4
 LOOKUP[4]=1, RESULT=5, ; curve 2 lookup from col 1 return value from col 5
 LOOKUP[5]=1, RESULT=6, ; curve 2 lookup from col 1 return value from col 6
 FAIL=0,0,0,
 INTERPOLATE=N
- Hourly Off Toll Ramp Capacities TOFFPCE(NEWAREA, lanes), not interpolated
 LOOKUP, NAME= TOFFPCE, LOOKUPI=10, ;(file 10)
 LOOKUP[1]=1, RESULT=2, ; curve 1 lookup from col 1 return value from col 2
 LOOKUP[2]=1, RESULT=3, ; curve 2 lookup from col 1 return value from col 3

LOOKUP[3]=1, RESULT=4, ; curve 2 lookup from col 1 return value from col 4
 LOOKUP[4]=1, RESULT=5, ; curve 2 lookup from col 1 return value from col 5
 LOOKUP[5]=1, RESULT=6, ; curve 2 lookup from col 1 return value from col 6
 FAIL=0,0,0,
 INTERPOLATE=N

- Calculate Capacities
- Links that are not centroid connectors and not transit-only (!(FTC2=51,52,49,59,69))
 - Links without manual capacity override (OVERRIDE=0)
 - Freeways /HOV/ Toll mainlines (FRWY=1)
 - CBD,CBD Fringe, Residential and OBD (NEWAREA=1-4)
 - CAPALL=FRWYPCE(1,NUM_LANES)
 - Rural (NEWAREA>44)
 - CAPALL=FRWYPCE(2,NUM_LANES)
 - More than 6 lanes (NUM_LANES>6)
 - CAPALL=FRWYPCE(1,6)
 - Toll plazas (TOLLPLAZA=1 & TOLLTYPE<>3)
 - CAPALL=1300
 - Peak Only and All-Day HOV Ramps (HOVPEAK=1 | HOVDAY=1)
 - CAPALL=4000
 - Uninterrupted Highways (UNINTRP=1)
 - CAPALL=HWYPCE(1,NUM_LANES)
 - More than 3 lanes (NUM_LANES>3)
 - CAPALL=HWYPCE(1,3)
 - Interrupted, No Low Speed Arterials, Non-Toll (FRWY=0 & RAMPS=0 & UNINTRP=0 & LOWSPD=0 & KTOLL=0)
 - Signalized (NUMSIG>0)
 - For (SIG_RATE>0 & SIG_RATE<=1.99)
 - CAPALL=ARTPCE(1,NUM_LANES)
 - For (SIG_RATE>1.99 & SIG_RATE<=4.5)
 - CAPALL=ARTPCE(2,NUM_LANES)
 - For (SIG_RATE>4.5)
 - Links not in primary CBD or OBD (NEWAREA=2-3 | NEWAREA=5)
 - CAPALL=ARTPCE(3,NUM_LANES)
 - Links in CBD/OBD
 - CAPALL=ARTPCE(4,NUM_LANES)
 - Other signalized
 - CAPALL=ARTPCE(5,NUM_LANES)
 - Unsignalized (Unsignalized, SEGID=0 and NUMSIG=0)
 - CAPALL=ARTPCE(6,NUM_LANES)
 - More than four lanes (NUM_LANES>4)
 - CAPALL=ARTPCE(1,4)
 - Low speed links (LOWSPD=1)
 - CBD and fringe (NEWAREA=1-2)

- $CAPALL = LOWPCE(1, NUM_LANES)$
- OBD (NEWAREA=4)
 - $CAPALL = LOWPCE(2, NUM_LANES)$
- Residential (NEWAREA=3)
 - $CAPALL = LOWPCE(3, NUM_LANES)$
- Rural (all others)
 - $CAPALL = LOWPCE(4, NUM_LANES)$
- More than 4 lanes ($NUM_LANES > 4$)
 - $CAPALL = LOWPCE(1, 4)$
 - print, list= 'LOWSPD', CAPALL(8.0); (report)
- Capacity Adjustments
- For Undivided Arterial/Non-State ($FRWY=0$ & $RAMPS=0$ & $UNINTRP=0$ & $HOV=0$ & $KTOLL=0$);
 - One-way links ($TWOWAY=0$)
 - $CAPALL = 1.20 * CAPALL$; (Add 20% to One-Way)
 - Left-Turn Adjustments for No-Left-Turn Bay for Interrupted links ($FRWY=0$ & $RAMPS=0$ & $UNINTRP=0$ & $HOV=0$ & $KTOLL=0$)
 - Links with more than one lane ($NUM_LANES > 1$)
 - Undivided ($DIVIDED=0$)
 - With left-turn bays ($LEFTTURN=1$)
 - $CAPALL = 0.95 * CAPALL$
 - Without left-turn bays ($LEFTTURN=0$)
 - $CAPALL = 0.75 * CAPALL$
 - Links with only one lane ($NUM_LANES > 1$)
 - Divided links ($DIVIDED=1$)
 - $CAPALL = 1.05 * CAPALL$
 - Undivided links ($DIVIDED=0$)
 - $CAPALL = 0.80 * CAPALL$
 - Non-HOV on/off ramps ($RAMPS=1$ & $HOV=0$ & $FRWY2FRWY=0$ & $TOLLPLAZA=0$)
 - Free links ($KTOLL=0$)
 - On ramps ($ON=1$)
 - Area type 1 (NEWAREA=1)
 - $CAPALL = ONPCE(1, NUM_LANES)$
 - Area type 2 (NEWAREA=2)
 - $CAPALL = ONPCE(2, NUM_LANES)$
 - Area type 3 (NEWAREA=3)
 - $CAPALL = ONPCE(3, NUM_LANES)$
 - Area type 4 (NEWAREA=4)
 - $CAPALL = ONPCE(4, NUM_LANES)$
 - Area type 5 (NEWAREA=5)
 - $CAPALL = ONPCE(5, NUM_LANES)$
 - More than 2 lanes ($NUM_LANES > 2$)
 - $CAPALL = ONPCE(1, 2)$
 - Off ramps ($OFF=1$)

- Area type 1 (NEWAREA=1)
 - CAPALL=OFFPCE(1,NUM_LANES)
 - Area type 2 (NEWAREA=2)
 - CAPALL= OFFPCE (2,NUM_LANES)
 - Area type 3 (NEWAREA=3)
 - CAPALL= OFFPCE (3,NUM_LANES)
 - Area type 4 (NEWAREA=4)
 - CAPALL= OFFPCE (4,NUM_LANES)
 - Area type 5 (NEWAREA=5)
 - CAPALL= OFFPCE (5,NUM_LANES)
 - More than 2 lanes (NUM_LANES>2)
 - CAPALL= OFFPCE(1,2)
- On Loop ramps (ONLOOP=1)
 - Area type 1 (NEWAREA=1)
 - CAPALL= LONPCE (1,NUM_LANES)
 - Area type 2 (NEWAREA=2)
 - CAPALL= LONPCE (2,NUM_LANES)
 - Area type 3 (NEWAREA=3)
 - CAPALL= LONPCE (3,NUM_LANES)
 - Area type 4 (NEWAREA=4)
 - CAPALL= L LONPCE (4,NUM_LANES)
 - Area type 5 (NEWAREA=5)
 - CAPALL= LONPCE (5,NUM_LANES)
 - More than 2 lanes (NUM_LANES>2)
 - CAPALL= LONPCE (1,2)
- Off Loop ramps (OFFLOOP=1)
 - Area type 1 (NEWAREA=1)
 - CAPALL= LOFFPCE (1,NUM_LANES)
 - Area type 2 (NEWAREA=2)
 - CAPALL= LOFFPCE (2,NUM_LANES)
 - Area type 3 (NEWAREA=3)
 - CAPALL= LOFFPCE (3,NUM_LANES)
 - Area type 4 (NEWAREA=4)
 - CAPALL= LOFFPCE (4,NUM_LANES)
 - Area type 5 (NEWAREA=5)
 - CAPALL= LOFFPCE (5,NUM_LANES)
 - More than 2 lanes (NUM_LANES>2)
 - CAPALL= LOFFPCE (1,2)
- Toll links (KTOLL=1)
 - Toll on ramps (TOLLON=1)
 - Area type 1 (NEWAREA=1)
 - CAPALL= TONPCE (1,NUM_LANES)
 - Area type 2 (NEWAREA=2)

- CAPALL= TONPCE (2,NUM_LANES)
 - Area type 3 (NEWAREA=3)
 - CAPALL= TONPCE (3,NUM_LANES)
 - Area type 4 (NEWAREA=4)
 - CAPALL= TONPCE (4,NUM_LANES)
 - Area type 5 (NEWAREA=5)
 - CAPALL= TONPCE (5,NUM_LANES)
 - More than 2 lanes (NUM_LANES>2)
 - CAPALL= TONPCE (1,2)
- Toll off ramps (TOLLOFF=1)
 - Area type 1 (NEWAREA=1)
 - CAPALL= TOFFPCE (1,NUM_LANES)
 - Area type 2 (NEWAREA=2)
 - CAPALL= TOFFPCE (2,NUM_LANES)
 - Area type 3 (NEWAREA=3)
 - CAPALL= TOFFPCE (3,NUM_LANES)
 - Area type 4 (NEWAREA=4)
 - CAPALL= TOFFPCE (4,NUM_LANES)
 - Area type 5 (NEWAREA=5)
 - CAPALL= TOFFPCE (5,NUM_LANES)
 - More than 2 lanes (NUM_LANES>2)
 - CAPALL= TOFFPCE (1,2)
- Set directional capacity per hour per lane: CAPACITY_LANE=CAPALL
- Set total directional capacity per hour: CAPACITY_HR=CAPACITY_LANE*NUM_LANES
- Set Centroid Connector Capacities
- For links that are centroid connectors and not transit-only (FTC1=50,59)
 - CAPALL=10000
 - CAPACITY_LANE=CAPALL ; Capacity per hour per lane
 - CAPACITY=CAPALL; this will override the CAPACITY field that exists on network
 - CAPACITY_HR=CAPACITY_LANE*NUM_LANES; Total Directional Hourly Capacity
- Print statement to allow check and debugging
- Print

list=A(7.0),B(7.0),FTC2(3.0),NUM_LANES(3.0),CAPALL(7.0),CAPACITY_LANE(7.0),CAPACITY_HR(7.0),CAPACITY(7.0)

(4.12) NTNET00A.S 'Create list of freeway on-ramp merge nodes'

VOYAGER NETWORK PROGRAM

This list supports NTNET00B.S and the HCM2010 speed adjustment needed for ramp merges for freeway on-ramps

Input Files

- LINKI=T4TEM4.NET (highway network)

Output Files

- PRINTO= rampend.csv (list of freeway on-ramp merge nodes)
 - Node number
 - myhot: 1 if HOT 0 if not HOT (all 0 for TCRPM4)
 - orient: 1=NB, 2=EB, 3=SB, 4=WB
 - myft: two-digit facility of the mainline link defined by the node

Declare Arrays to be Used Later

- ARRAY ismrg=999999, isramp=999999, xc=999999, yc=999999, orient=999999, myhot=999999, myft=999999

Process Each Node (NODEMERGE) the following logic is applied to each node:

- xc[n]=x; store node coordinates
- yc[n]=y

Process Each Link (LINKMERGE) the following logic is applied to each link:

- Highway links that are freeways or turnpike (FTC2=11,12,91,92)
 - ismrg[A]=1; include all freeway nodes
 - ismrg[B]=1
 - myft[a]=ftc2; store two-digit facility type
 - myft[b]=ftc2
 - ; get orientation from anode to bnode using coordinates
 - IF(xc[a]<xc[b])
 - _ew=1; toward the east
 - ELSE
 - _ew=0
 - IF(yc[a]<yc[b])
 - _ns=1; toward the north
 - ELSE
 - _ns=0
 - _dx=abs(xc[a]-xc[b])
 - _dy=abs(yc[a]-yc[b])
 - IF(_dx > _dy); either EW or WE (is change greater NS or EW?)
 - IF(_ew=1)
 - _mydir=2; eastbound

- ELSE
 - _mydir=4; westbound
 - ELSE ; either NS or SN
 - IF(_ns=1)
 - _mydir=1; northbound
 - ELSE
 - _mydir=3; southbound
 - orient[a]=_mydir; orientation is the same for both nodes
 - orient[b]=_mydir
 - IF(HOT=1-98); is this a HOT lane facility
 - myhot[a]=1
 - myhot[b]=1
 - ELSE
 - myhot[a]=0
 - myhot[b]=0
- On-ramps (FTC2=71-79,93,94)
 - isramp[a]=1
 - isramp[b]=1

PHASE=SUMMARY (Summarize) After all links are processed:

- LOOP _k=1,99999; (check all nodes and print merge nodes)
 - if nodes on both freeway and ramp links and orientation is known
 - IF((ismrg[_k]*isramp[_k] = 1) & (orient[_k]>0))
 - Print node, hotflag, orientation, and facility type
 - print csv=T list=_k(6.0),myhot[_k](2.0),orient[_k](3.0),myft[_k](3.0) printo=1
- ENDLOOP



(4.13) NTNET00B.S 'Speed Calculator'

VOYAGER NETWORK PROGRAM

Calculates and adjusts free flow speed for all links in the network

Input Files

- LINKI=T4ATEM4.NET (highway network)
- NODEI[2]=SIGS1.DBF (highway network signalized nodes)
- NODEI[3]=RAMPEND.CSV (list of on-ramp merge nodes)
 - VAR=n,mtyp,orient,myft
- LOOKUPI = signal segment data
 1. SEGMENT.CSV – signal segment data
 2. LASTLINKNODES.PRN – signal segment terminal link node numbers

Output Files

- LINKO= SEGSCYCLE.DBF (list of links and signal cycle lengths)
 - INCLUDE=A,B,CYCLELENGTH
- NETO=T4E.NET (Network with speeds and capacities)

Define Lookup Tables

- Signalized segments SEGDATA (#tab,SEGID)
LOOKUPI=1 LIST=N NAME=SEGDATA,
LOOKUP[1]=1, RESULT=2, ; curve 1 lookup from col 1 return SEG LENGTH
LOOKUP[2]=1, RESULT=3, ; curve 1 lookup from col 1 return NUMSIG
LOOKUP[3]=1, RESULT=4, ; curve 1 lookup from col 1 return FT1SEGS
LOOKUP[4]=1, RESULT=5, ; curve 1 lookup from col 1 return CYCLE
LOOKUP[5]=1, RESULT=6, ; curve 1 lookup from col 1 return GC
LOOKUP[6]=1, RESULT=7, ; curve 1 lookup from col 1 return DF
LOOKUP[7]=1, RESULT=8, ; curve 1 lookup from col 1 return DD
INTERPOLATE =N; not interpolated
- Last node in signal segment LASTNODE (1, BNODE)
LOOKUP, NAME= LASTNODE, LOOKUPI=2, ;(file 2)
LOOKUP[1]=1, RESULT=2, ; curve 1 lookup from BNODE col 1 return 1 from col 2 if last
FAIL=0,0,0,
INTERPOLATE=N ; (not interpolated)

Declare Arrays to be Used Later

- ARRAY xc=999999,yc=999999,oc=999999,ftn=999999

Process Each Node (NODEMERGE) the following logic is applied to each node:

- xc[n]=x/5280; store node coordinates in miles
- yc[n]=y/5280; store node coordinates in miles
- oc[n]=orient; mainline orientation at merge node from rampends: 1=NB,2=EB,3=SB,4=WB
- _nmax=n; saves max node number (assume in ascending sort)
- ftn[n]=myft; save facility type for each node

Process Each Link (LINKMERGE) the following logic is applied to each link:

- Highway links that are not centroid connectors or transit-only (FTC2<>49 | FTC2<>50-59 | FTC2<>69)
 - $_slen = \text{SEGDATA}(1, \text{SEGID})$; segment length from curve 1 for the current link SEGID
 - $_nsig = \text{SEGDATA}(2, \text{SEGID})$; number of signals from curve 2 for the current link SEGID
 - $_DF = \text{SEGDATA}(6, \text{SEGID})$; signal coordination factor DF from curve 6 for the current link SEGID
 - $_CYCLE = \text{LI.1.CYCLELENGTH}$; signal cycle length added to links in earlier steps
 - $_LSTNOD = \text{LASTNODE}(1, \text{LI.1.B})$; last node of the signal segment
 - If cycle length is greater than zero ($_CYCLE > 0$)
 - $_LOSTTIME = 4 / _CYCLE$; estimated lost time every cycle
 - ELSE $_LOSTTIME = 0$
 - IF ($\text{LI.1.GC_RATIO} > _LOSTTIME$)
 - $_GCM = 1 - \text{LI.1.GC_RATIO} - _LOSTTIME$; Subtract lost time of 4 secs per approach ($4 / _CYCLE =$ lost time ratios with respect to Cycle length)
 - ELSE $_GCM = 1 - \text{LI.1.GC_RATIO}$
 - IF ($_GCM < 1$)
 - $_DD = 0.5 * _DF * _CYCLE * _GCM^2$; Average delay per signal (seconds)
 - ELSE
 - $_DD = \text{SEGDATA}(7, \text{SEGID})$
 - $_secphr = 1.0 / 3600$; hours per second
 - $_FT1 = \text{INT}(0.1 * \text{FTC2})$; Compute one-digit facility type code from two-digit code
 - Speed adjustment factors
 - $_fwyfac = 1.0$
 - $_rampfac = 0.8$
 - $_fac21 = 1.0$
 - $_fac61 = 1.0$
 - Non-Centroid and non-Transit-Only optional links ($(_FT1 < 5) \& (!(\text{FTC2} = 49, 59, 69))$)
 - Freeway & HOV (FTC2=11-12,81-82)
 - $\text{FFSPD} = (12.56418 + 0.64058 * \text{POSTED_SPD} + 0.00340 * (\text{POSTED_SPD})^2) * _fwyfac$
 - Mainline Toll Links (FTC2=91-92)
 - $\text{FFSPD} = (12.56418 + 0.64058 * \text{POSTED_SPD} + 0.00340 * (\text{POSTED_SPD})^2) * _fwyfac$
 - Uninterrupted Roadways (FTC2=21)
 - $\text{FFSPD} = (10.10 + 0.65735 * \text{POSTED_SPD} + 0.00348 * (\text{POSTED_SPD})^2) * _fac21$
 - High Speed Arterials (FTC2=41)
 - $\text{FFSPD} = (10.10 + 0.65735 * \text{POSTED_SPD} + 0.00348 * (\text{POSTED_SPD})^2)$
 - Low Speed Collectors (FTC2=61)
 - $\text{FFSPD} = (10.10 + 0.65735 * \text{POSTED_SPD} + 0.00348 * (\text{POSTED_SPD})^2) * _fac61$
 - Ramps-Freeway (FTC2=71-75) ;
 - $\text{FFSPD} = (11.33209 + 0.64896 * \text{POSTED_SPD} + 0.00344 * (\text{POSTED_SPD})^2) * _rampfac$; slow down ramps by 20%
 - Ramps-Toll Facilities (FTC2=93,94)

- $FFSPD = (11.33209 + 0.64896 * POSTED_SPD + 0.00344 * (POSTED_SPD)^2) * _rampfac$; slow down ramps by 20%
- HOV Slip Ramps (FTC2=83-86) ;
 - $FFSPD = (11.33209 + 0.64896 * POSTED_SPD + 0.00344 * (POSTED_SPD)^2)$
- All other roads (for example Toll plaza-95) but not cencons
 - $FFSPD = (11.33209 + 0.64896 * POSTED_SPD + 0.00344 * (POSTED_SPD)^2)$
- Signalized links (li.1.SEGID <> 0 & _LSTNOD <> 1)
 - $FFSPD = _slen / ((_slen / FFSPD) + _nsig * _DD * _secphr)$; travel speed plus delay
 - IF (FFSPD=0)
 - _ecount=_ecount+1; count links with zero speed and print error information
 - PRINT LIST='FFSPD computed = ',FFSPD
 - PRINT LIST='SEGID= ',SEGID
 - PRINT LIST='slen= ',_slen
 - PRINT LIST='nsig= ',_nsig
 - PRINT LIST='DD= ',_DD
 - PRINT LIST='secphr= ',_secphr
 - _icount=_icount+1; count all links
- ELSE; (count unsignalized links)
 - _ffcount=_ffcount+1
- ELSE (cencon and transit only)
 - $FFSPD = POSTED_SPD$

Apply Ramp Merge Adjustments

Note: this adjustment impacts all FWY & TPK links according to HCM 2010 formula and imposes a penalty for frequent merges.

- _TRD=0; initialize ramp density
- MERGEPEN=0; initialize merge penalty
- IF(FTC2=11,12,81,82,91,92 & POSTED_SPD>45) don't include causeway toll booths
- ; estimate every link orientation so ramp density can be correctly counted
- IF(xc[a]<xc[b])
 - _ew=1; toward the east
- ELSE
 - _ew=0
- IF(yc[a]<yc[b])
 - _ns=1; toward the north
- ELSE
 - _ns=0
- _dx=abs(xc[a]-xc[b])
- _dy=abs(yc[a]-yc[b])
- IF(_dx > _dy); either EW or WE
 - IF(_ew=1)
 - mydir=2; eastbound

- ELSE
 - mydir=4; westbound
- ELSE ; either NS or SN
 - IF(_ns=1)
 - mydir=1; northbound
 - ELSE
 - mydir=3; southbound
- ; midpoint of the link
- $_mx = 0.5 * (xc[a] + xc[b])$; get the midpoint of the link
- $_my = 0.5 * (yc[a] + yc[b])$
- $_kk = 0$
- $_ftc1 = _FT1$; link one-digit facility type
- LOOP $_k = 1, _nmax$; loop through all nodes
 - $_ftn1 = \text{int}(\text{ftn}[_k] / 10)$; one digit facility type for the node
 - IF($_ftc1 = _ftn1$); link and node facility type are the same
 - $_ok = 1$
 - ELSEIF($_ftc1 = 8 \ \& \ _ftn1 = 1$); allows merge delays to accrue to HOV lanes
 - $_ok = 1$
 - ELSE; node and link facility types don't match
 - $_ok = 0$
 - IF($HOT = 1-98$); no merge delays on HOT lanes
 - $_ok = 0$

Link and potential ramp end must be in the same direction
and must be same FTC2 so close TPK and I-95 are not confused

- IF($(oc[_k] = mydir) \ \& \ _ok = 1 \ \& \ \text{ftn}[_k] > 0$)

Find the distance between the ramp end merge node and each link with matching orientation, and verify the ramp lies close to an A-B straight LINE. Deal with squares to avoid sqrt calculation. $_dv$ is variation from the LINE and $_d2$ is the square of the distance from the midpoint of the link and the ramp end. All Pythagorean.

- $_denom = ((xc[b] - xc[a])^2 + (yc[b] - yc[a])^2)^{0.5}$
- $_dv = \text{abs}((xc[b] - xc[a]) * (yc[a] - yc[_k]) - (xc[a] - xc[_k]) * (yc[b] - yc[a])) / _denom$; distance between ramp end and mainline LINE
- $_dx = (xc[_k] - _mx)$
- $_dy = (yc[_k] - _my)$
- $_d2 = (_dx^2 + _dy^2)$
- IF($_dv < 0.25 \ \& \ _d2 \leq 9$); close enough to count?
 - $_kk = _kk + 1$; 3 miles squared = 9 and no more than 0.25 mile off A-B line
 - print list=a(6.0),b(6.0), $_k$ (6.0), $_kk$ (6.0); print to debug
- ENDLOOP
- $_TRD = _kk / 6$; ramp density along a 6 mile segment as specified by HCM2010. This is 3 miles in each direction, upstream and downstream.
- IF($FFSPD > 0$)

Here we are starting with the SERPM 2nd order equation FFSPD as calculated earlier, rather than 75.4 and specified in HCM 2010. Apply only 80% of penalty to HOV's because they are at least a lane away from the entry ramps.

- IF(_ftc1=8); HOV's
 - _hvfac=0.80; 80% as noted above
- ELSE
 - _hvfac=1.0
- MERGEPEN=_hvfac*3.22*_TRD^0.84; the merge penalty in mph.
- FFSPD=FFSPD-MERGEPEN; HCM2010 method to account for ramp density

Apply adjustment factors and convert to Integer speeds for use in FSUTMS

- _ADJF=1.0
- IF (LI.1.POSTED_SPD <> 0)
 - IF (LI.1.HOT = 0); apply if not a HOT lane
 - SPD1=round(100*FFSPD*_ADJF)/100
 - ELSE; do not adjust HOT lanes speed
 - SPD1=round(100*FFSPD)/100
- POSTEDSPEED=POSTED_SPD
- IF (LI.1.POSTED_SPD<>0); links with posted speeds – should be all of them
 - FREEFLOWSPEED=SPD1
- ELSE
 - FREEFLOWSPEED=FFSPD

Override FFSPD for any Link Distance less than 0.005 mile to trap errors

- IF (LI.1.DISTANCE<=0.005) FFSPD=POSTED_SPD
- IF (LI.1.DISTANCE<=0.005) FREEFLOWSPEED=POSTED_SPD
- IF (LI.1.DISTANCE>0.005)
 - IF (POSTED_SPD>0) POSTEDTIME=60*DISTANCE/POSTED_SPD

HOT lane speed advantage added for SERPM7

- IF (HOT=1-98) FREEFLOWSPEED=FREEFLOWSPEED + {HOTSPDADD}; use **Key**
- IF (FREEFLOWSPEED>0) FREEFLOWTIME=60*DISTANCE/FREEFLOWSPEED
- IF (FREEFLOWTIME>0)
 - TIME=FREEFLOWTIME
 - TIME2=FREEFLOWTIME
- ELSE; distance(0.005mile) & time(0.01min) equivalent to 30mph speed
 - POSTEDTIME=0.01
 - FREEFLOWTIME=0.01
 - TIME=0.01
 - TIME2=0.01

Compute SPEED and SPPED2 from TIME and TIME2

- if(!(FTC2=49,59,69)); Highway Network Links Only
 - IF (TIME>0)
 - SPEED=DISTANCE/TIME*60
 - IF(DISTANCE<0.005) SPEED=30

- ELSE
 - SPEED=POSTEDSPEED
 - TIME=FREEFLOWTIME
- IF (TIME2>0)
 - SPEED2=DISTANCE/TIME2*60
 - IF(DISTANCE<0.005) SPEED2=30
- ELSE
 - SPEED2=POSTEDSPEED
 - TIME2=FREEFLOWTIME
- IF (FTC2=95); For Toll Plaza Links
 - TIME=0
 - TIME2=0

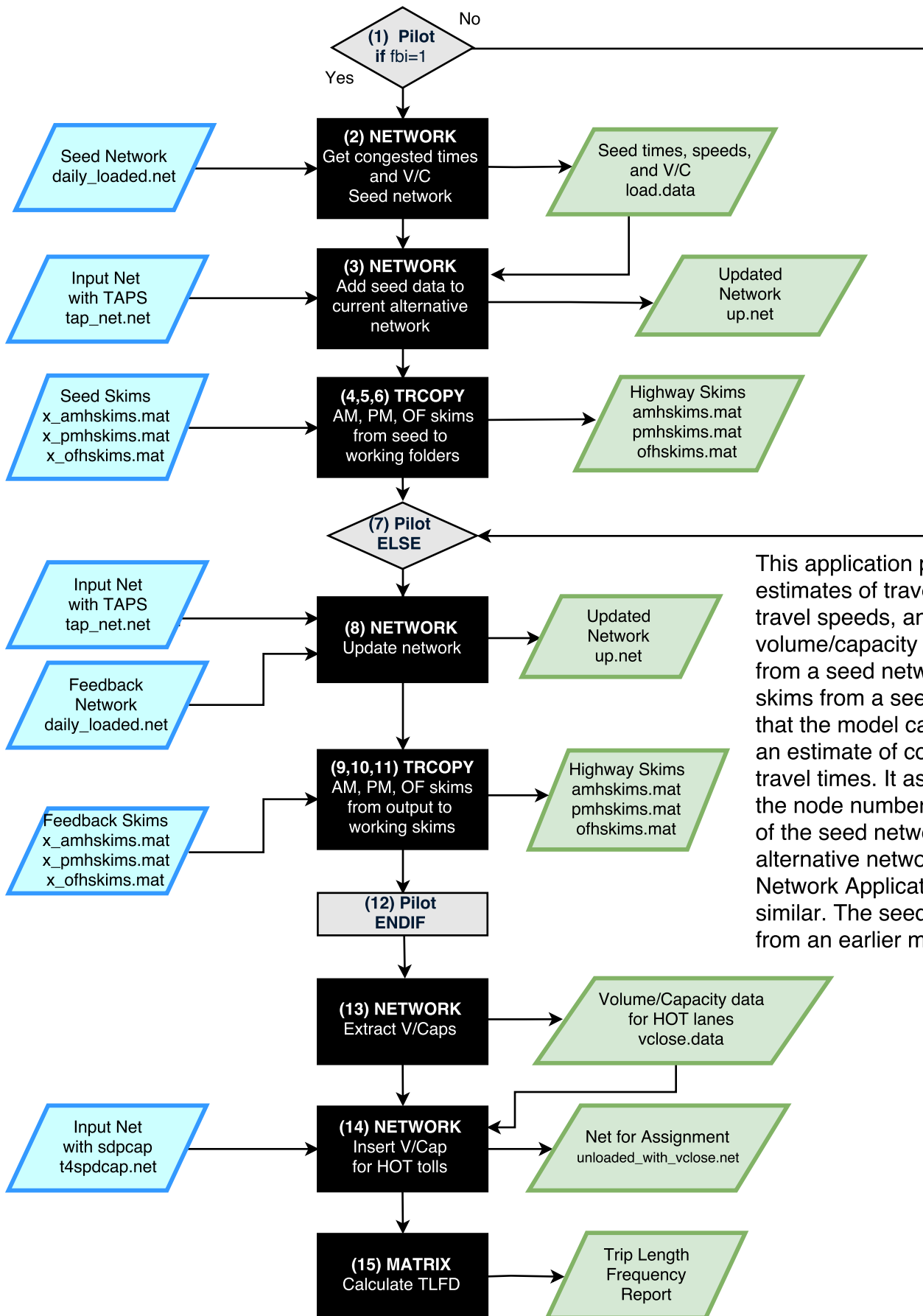
Set Speed and Time for Transit-Only Optional Links

- IF (FTC2=49,59,69 & TMODE>0); transit-only links
 - SPEED=TSPEED
 - SPEED2=TSPEED
 - TIME=TTIME
 - TIME2=TTIME
 - FREEFLOWTIME=TTIME

PHASE=SUMMARY (Summarize) After all links are processed:

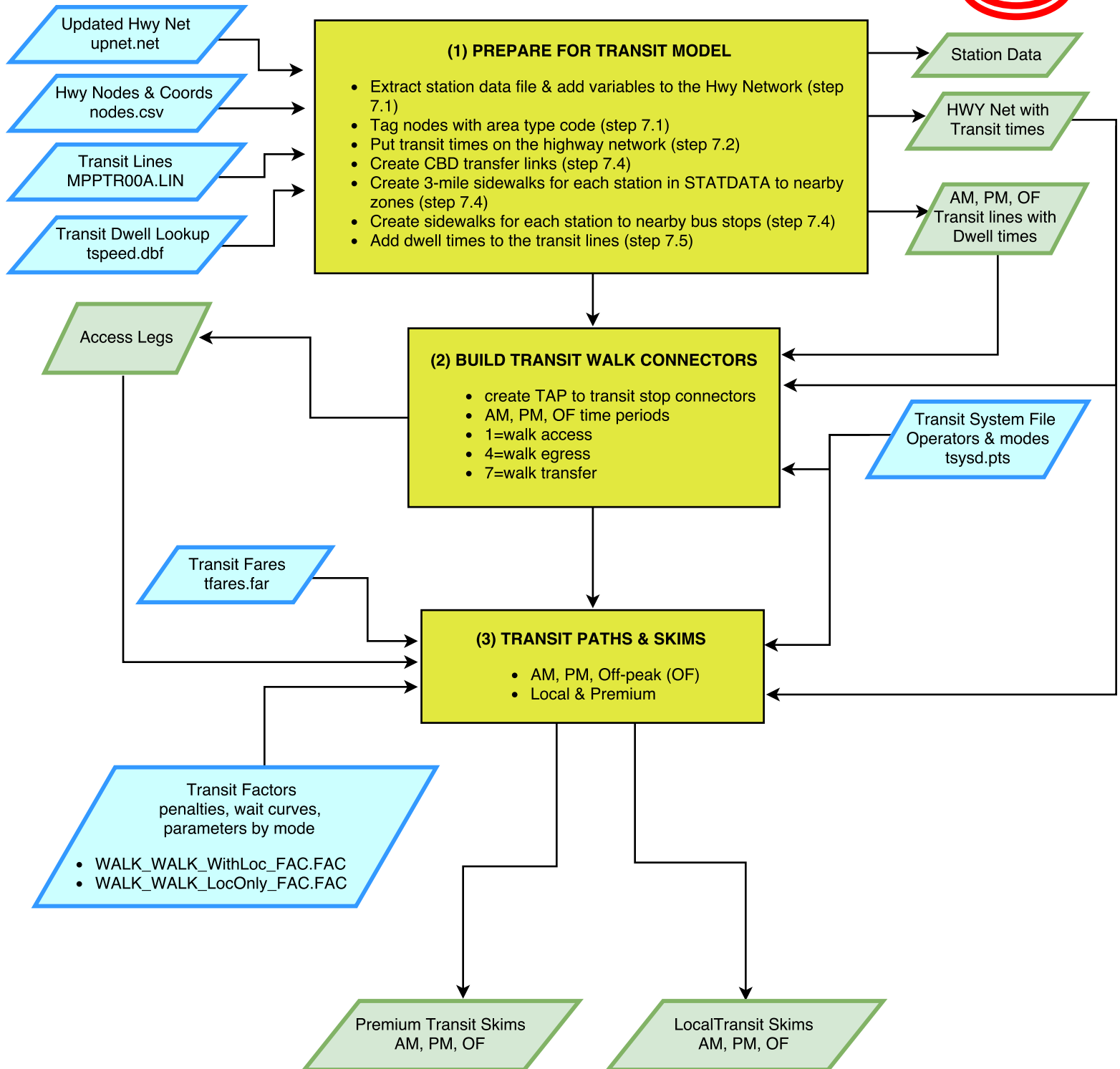
- IF (_ecount>0) PRINT LIST=_ecount,' links with segids were assigned speed=0'
-
- PRINT LIST=_ffcount,' Free-flow links'
- PRINT LIST=_icount,' intersection controlled links'

(6) Netlink.app (Create/Update Highway Network and Skims)



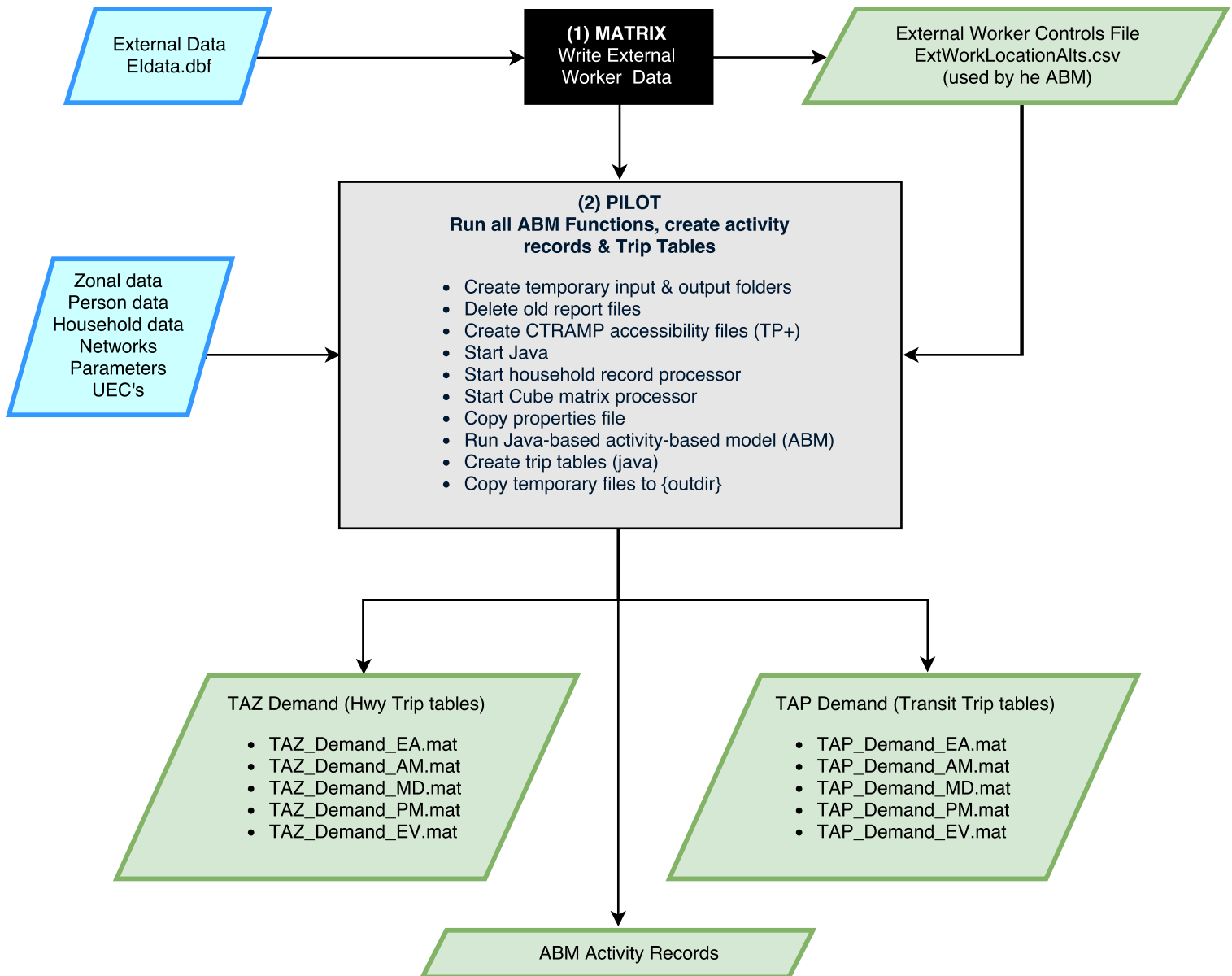
This application provides initial estimates of travel times, travel speeds, and network volume/capacity ratios (V/C's) from a seed network, and skims from a seed matrix, so that the model can begin with an estimate of congested travel times. It assumes that the node numbering scheme of the seed network and the alternative network from the Network Application are very similar. The seed files come from an earlier model run.

(7) Transit Network, Paths & Skims



This application builds AM, PM, and off-peak transit networks, connectors, paths and skims. Step 2 specifies how Transit Access Points (TAPS) are connected to the network. It is important to note that in CT-RAMP models, trip tables and skims are TAP-to-TAP, not TAZ-to-TAZ. TAPS are constructed earlier in the Network Application. Skims are developed for local-only transit service, and premium transit service, for each time period. Skims include out-of-vehicle travel time, in-vehicle travel time, initial wait time, number of transfers, transfer wait time, and fares.

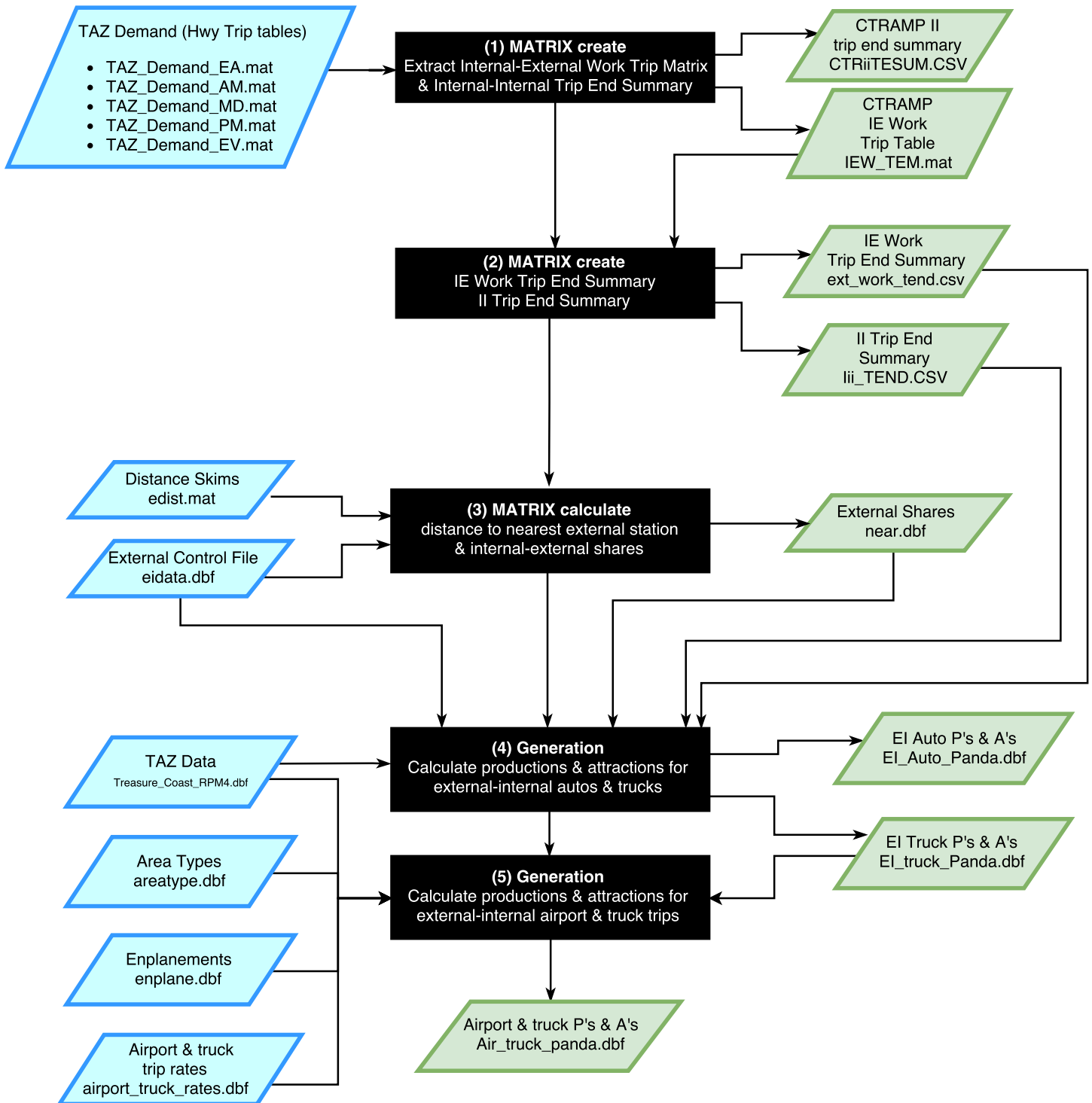
(8) RUN CT-RAMP



The initial Matrix program reads the Eldata control file, and converts it to the format needed by the external worker model.

The Pilot program runs all of the scripts needed to create the CT-RAMP accessibility files, runs the activity-based model (ABM), and converts the output trip logs to trip tables that can be assigned by Cube programs to highway and transit networks. Note that transit network centroids are transit access points (TAP's), while highway network centroids are TAZ's. The ABM uses all available computer memory and threads, and is implemented through two Java servers, one to read Cube skims and provide them to the Java-based ABM, and another to process household records to produce daily travel logs for every synthetic person in the region.

(9) Non-ABM Trip Generation

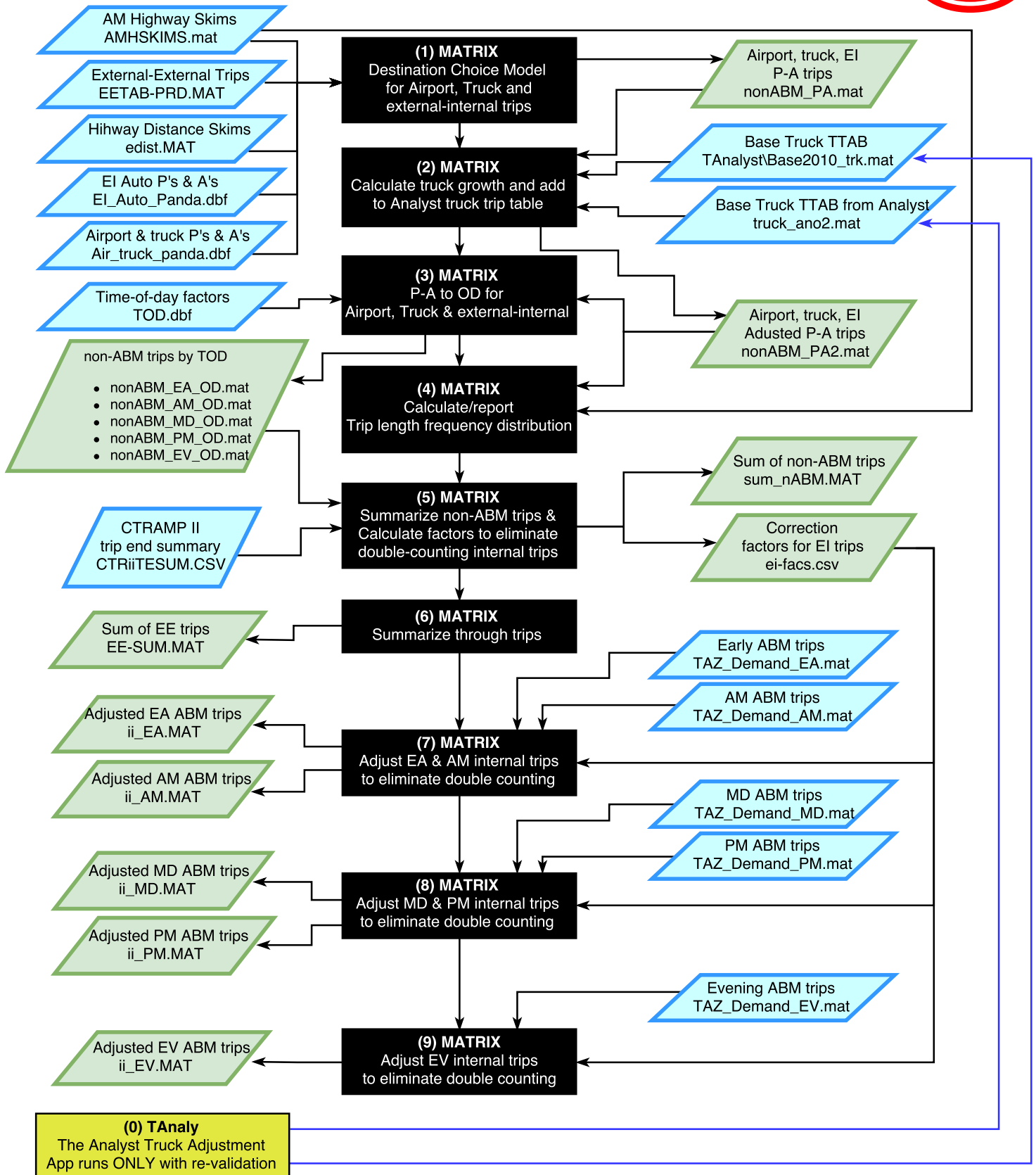


The ABM creates all trips made by persons living in households inside the region that have both ends inside the region. All other trips, except through (E-E) trips, which eventually appear on the Treasure Coast highway and transit networks, are generated in this application. These trips are internal-external (IE) and external-internal (EI) auto trips, airport trips, and truck trips.

Additionally, this step creates a trip end summary of the ABM external worker model trips (people living inside the Treasure Coast region but working in other counties), which is used later in the non-ABM distribution step to prevent double counting IE trips.

(10) Non-ABM Trip Distribution

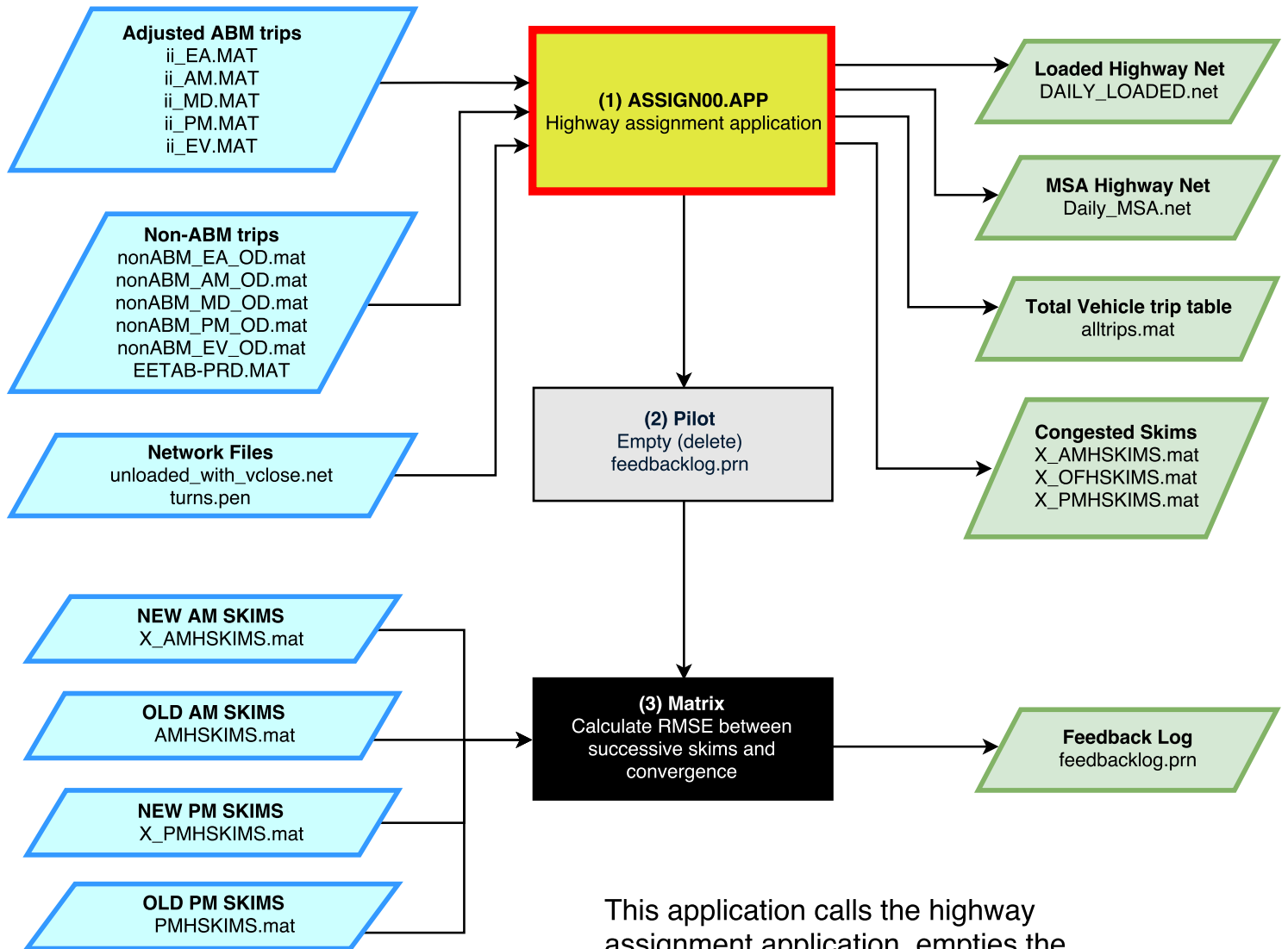
BACK



(The TAnalyst application creates the analyst and base year truck trip tables. It is run ONLY during validation)

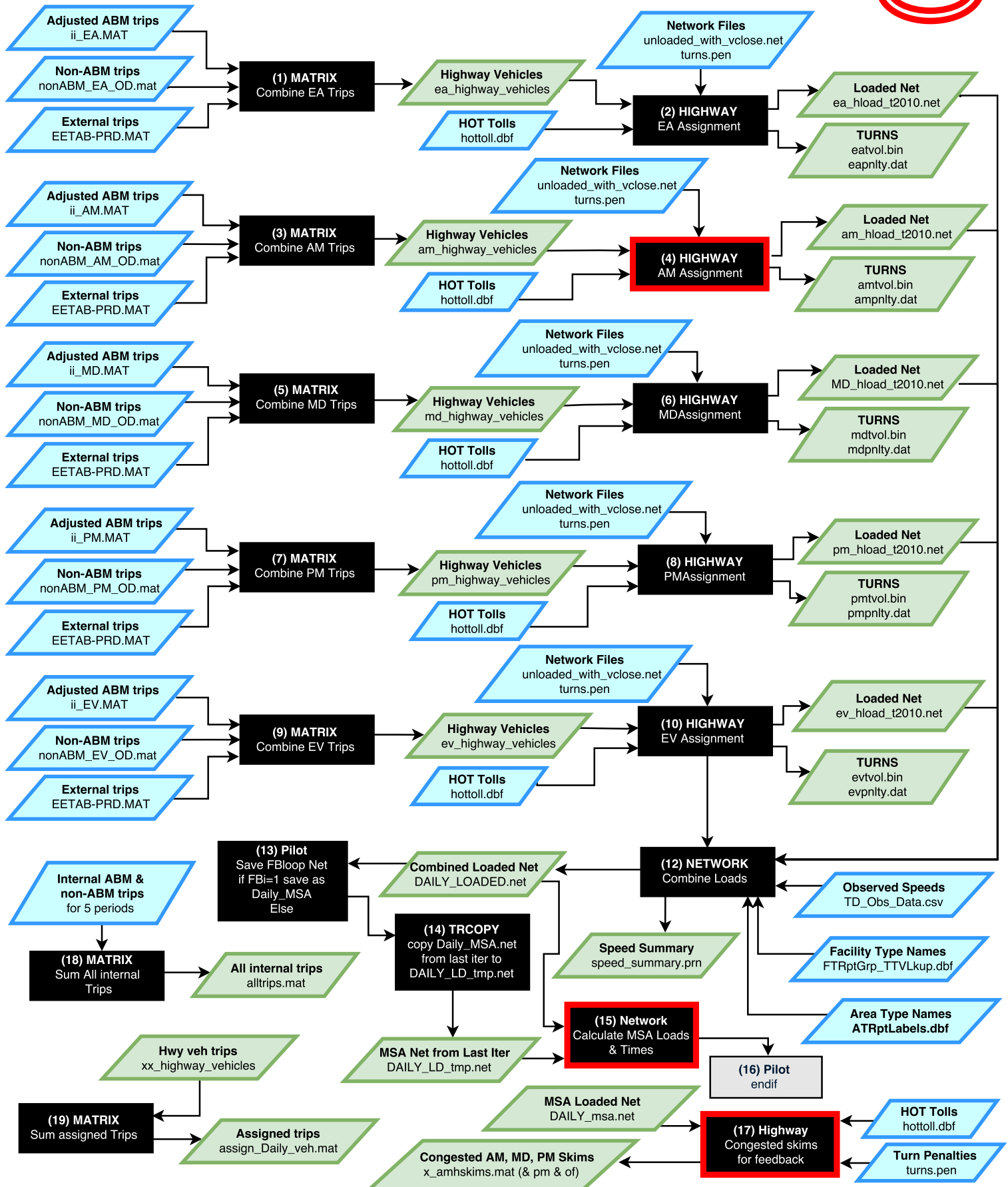
This application distributes the trips generated in the non-ABM trip generation step. It also uses a Cube Analyst-based trip table to create an incremental truck trip table, which minimizes differences between truck counts and model volumes in the base year. It also adjusts the ABM trip table so that external worker trips and external-internal trips are not double counted.

(11) Assignment



This application calls the highway assignment application, empties the feedback log file, calculates the %RMSE between the current highway skim matrix and the prior matrix, and writes a summary of the degree of convergence to the feedback log.

(11.1) Highway Assignment



(11.1.4) HAHWY00B.S 'AM Pk Assignment'

VOYAGER HIGHWAY PROGRAM

Input Files

- TURNPENI=Turn Penalties
- NETI=Highway Network
- MATI[1]=Highway Trip Tables
- LOOKUPI=Managed lanes tolls
- read FILE = "C:\TCRPM4\CUBE\LOAD_TCRPM.INC" Script common to all time periods

Output Files

- TURNPEN0=Output Turn Penalties (information only)
- TURNVOLO[1]= Turning volumes for selected intersections
- NETO=Loaded Network

Initialize

- Use all available Cluster nodes
- Set parameters
 - Zones from **KEY**
 - Maxiters from **KEY** (300)
 - COMBINE: equilibrium method =EQU
 - ENHANCE: Equilibrium subparameter from ={Enhance}=1
 - BIFWVERSION=602, use Cube version 602 method so turning reporting is correct
 - RELATIVEGAP: convergence criterion from **KEY** (0.0001)
- Define hottoll lookup values (htrate). Currently one system. Toll as a function of V/C. Rate is cost/mile in dollars.
- Define deceleration curve (DECEL), standard FSUTMS values for toll booths

Process Each Link (LINKREAD)

- T0 (time) is LI.TIME
- C (capacity) is LI.CAPACITY*[number of capacity hours in the period as set by **KEY**]
- BPR Coefficient for congested time read from LI.BPRCOEFFICIENT
- Initialize HOTVC from LI.AM_VCLOSE, used to define initial hot lanes tolls
- Set LINKCLASS from FTC2 and toll acceleration and deceleration codes
 - 1=no toll
 - 2=toll plaza
 - 3=Toll acceleration link
 - 4=Toll deceleration link
 - 5=Transit non-highway optional link (usually bus-only)
 - 9=any other link
- Set toll plaza times (FTC2=95)
 - T0=0
 - LW.SVCTIME from LI.SVCMINUTES and LI.SVCSECONDS
- Set ctoll values from LI.RCTOLL and period specific differences (CTOLL is inverse of VOT in min/dollar)

- Set express lane tolls from htrate lookup function and HOTVC
- Set total link toll (LW.TOLL_PKxx) by occupancy from fixed toll (LI.CARTOLL) plus htrate times link length
- Set impedance (GCOST) as the sum of:
 - T0 (time)
 - Toll service plaza time
 - Toll in dollars*RCTOLLxx
- Set up assignment groups
 - Group 1 includes express lanes and HOV ramps.
 - Group 3 includes toll lanes and express lanes
 - Group 4 includes toll lanes, express lanes and HOV ramps
 - Group 9 includes non-highway links, reversible links not in operation in the assignment period, and dummy HOT ramps (may be coded for testing).

Process each trip interchange (ILOOP)

- Define working matrices holding trip tables (MW)
 - 1=Single occupant free (SOV_GP)
 - 2=Single occupant toll (SOV_Pay)
 - 3=Two occupant shared ride free (SR2_GP)
 - 4=Two occupant shared ride HOV (SR2_HOV)
 - 5=Two occupant shared ride toll (SR2_Pay)
 - 6=Three or more occupant shared ride free (SR3_GP)
 - 7=Three or more occupant shared ride HOV (SR3_HOV)
 - 8=Three or more occupant shared ride toll (SR3_Pay)
 - 9=Truck vehicles
 - 10=All external (EE+IE+EI)
- “Read” common included assignment script (**LOAD_TCRPM.INC**), and pass the selected link **KEY** {sellink} – see description to follow. This section contains the “PATHLOAD” statements.

ADJUST PHASE

- APPLIES THE CAPACITY-RESTRAINT PROCEDURE
- Set volume for volume delay (V). This is the sum of autos and truck PCE's. {PCE} is a **KEY**.
- Set volume delay functions for linkclasses 1, 3, 4, and 6. All others are simply T0. The volume delay functions use the BPR form and the values for coefficients and exponents coded on the links.
- Update express lane costs, tolls, and GCOST based on the V/C from this assignment.

CONVERGE PHASE

- Require at least 6 assignment feedback iterations.
- Require relative gap convergence to be met in three consecutive iterations.
- Stop if the relative gap no longer changes even if the relative gap threshold is not met.

COMMON INCLUDED ASSIGNMENT SCRIPT (LOAD_TCRPM.INC) READ IN ILOOP

Coded this way to ensure that all periods use the same basic assignment structure.

- Load General Purpose links, excluding Group 4 (toll lanes, express lanes, HOV ramps).
 - 1=Drive alone, general purpose

- 3= Shared ride 2, general purpose
 - 6= Shared ride 3+, general purpose
- Load HOV links, excluding Group 3 (toll lanes and express lanes).
 - 4= Shared ride 2 toll and express
 - 7= Shared ride 3+ toll and express
- Load “pay” links, either toll or express, excluding only Group 9 (inactive=non-highway links, reversible links not in operation in the assignment period, and dummy HOT ramps).
 - 2= Drive alone, toll or express
 - 5=Shared ride 2, toll or express
 - 8=Shared ride 3+, toll or express
- Load Trucks and externals (EE+IE/EI)
 - 9=Trucks excluded from Groups 1 & 9 (HOT, HOV and inactive links).
 - 10=Externals – allowed in HOV and express.
- If this is a selected link/zone assignment, put selected loads in volume group fields 11-20, corresponding to the standard loads in groups 1-10.

Treasure Coast RPM PATHLOAD and Selected Link/Selected Zone Method

Overview

In TCRPM4, the pathload statements, which govern the way that trip tables are assigned to the highway network, are kept in a file called “LOAD_TCRPM.INC”, which resides in “C:\TCRPM4\CUBE” with all other TCRPM model scripts. This is done as a matter of programming convenience, as all time periods use the same set of code. Using this method, if an adjustment is made to the highway assignment pathload statements, changes made to “LOAD_TCRPM.INC” will appear in the code for all time periods. Cube does not support subroutines and sub-macros directly, but the common code is called in the assignment scripts with the following code:

```
myrun={RUNSELLINK}  
read FILE = "C:\TCRPM4\CUBE\LOAD_TCRPM.INC",  
  'pairs'='{sellink}'
```

This code has the effect of including “LOAD_TCRPM.INC” in the calling script, and passes the values of “myrun”, and “pairs” to the included code. My run is set to the value of the Cube Key {RUNSELLINK}, which is 0 if the “Run Selected Link Assignment?” checkbox on the interface is not checked, and 1 if it is checked. Thus, if “myrun” is 1, the selected link assignment will be run. Otherwise, the standard assignment will be run. The variable “pairs” gets its value from the {sellink} Cube Key, which is gotten from the “Enter Selected Link or Node Specification” in the user interface. Either selected links or selected nodes, including TAZ’s, can be specified.

Standard, Non-Selected Link Assignment

The standard assignment, for each time period, assigns trip tables to ten volume sets. Additionally, volume sets 11-20 are created, but they will be empty. In TCRPM4, all volume sets posted on the network are in vehicles, not passenger car equivalents. The volume sets appear individually in the network. Sets labeled as:

- V1_1, V2_1 ... V10_1 are one-way volumes.
- V1T_1, V2T_1 ... V10T_1 are two-way volumes.
- V_1 is the sum of all (1-20) one-way volumes.
- VT_1 is the sum of all (1-20) two-way volumes.

The ten volume sets are as follows:

- Volume sets 1, 3, and 6 are general purpose lanes, excluding toll lanes, express lanes, HOV ramps, and non-highway and other prohibited links. Set 1 contains drive-alone vehicles, set 3 contains 2-person vehicles, and set 6 contains 3 or more person vehicles.
- Volume sets 4 and 7 are HOV lanes, excluding toll and HOT lanes, and non-highway and other prohibited links. Set 4 contains 2-person vehicles, and set 7 contains 3 or more person vehicles.

- Volume sets 2, 5, and 8 are pay lanes (Turnpike or HOT), excluding non-highway and other prohibited links. Set 2 contains drive-alone vehicles, set 5 contains 2-person vehicles, and set 8 contains 3 or more person vehicles.
- Volume sets 9 and 10 are trucks and external (EE + EI/IE), excluding non-highway and other prohibited links. Set 9 contains trucks and set 10 contains external vehicles.

Later, after the assignments for all time periods have been run, the volumes are combined and are posted with more intuitive names.

Selected Link Assignment

A selected link, node, or zone assignment is run if the “Run Selected Link Assignment?” checkbox on the interface is checked, and there are values entered in “Enter Selected Link or Node Specification” in the user interface.

The structure of the load sets is the same as for the standard assignment, except that addition to sets 1-10, which contains loads for all links, sets 11-20 contain loads for the selected links. The offset between the total load number and selected load number is 10. For example, set 1 contains all general-purpose lane drive-alone vehicles, and set 11 contains general purpose lane drive-alone vehicles that meet the selected link specification.

Selected links are documented extensively in the Cube/Voyager documentation. TCRPM4 is set up to automatically use the **Selectlink** features, as entered in the “Enter Selected Link or Node Specification”. The following discussion has been excerpted from the Cube “Highway” documentation, and demonstrates the flexibility of the **Selectlink** specification (expression):

The expression is used to determine if the I-J path uses certain facilities. The expression must be enclosed within parenthesis (the script already contains the outer parenthesis), and can contain any of the following variables:

- **A** - The A-node of a link; only useful to select links with a path begins at A.
- **B** - The B-node of a link; only useful to select links with a path ends at B.
- **N** - A node that must be used in the path.
- **L** - A link that must be used in the path. Links are coded as A-B (directional) or A-B* (non-directional – link may be traversed in either direction).

*Any of these variables can have multiple values specified for them. Nodes (**A**, **B**, **N**) can be specified as single values and/or as ranges; **L** can have multiple links specified. When the multiple value specification is used, the implication is a logical OR amongst the values. Example: N=100,105,200-205 means if N is 100, or if N is 105, or if N is a value between 200 and 205, inclusive. There is no limitation on the number of nodes.*

The selection is processed by examining each link/node in the path, on both an individual link and a total path basis. The expression is processed from left to right. Each link/node is processed, and as long as it does not fail any conditions, it is considered as a candidate for processing in the total path. In the total path basis, the link/node is considered in conjunction with other links/nodes in the path. Here are some examples:

(A=1) Node 1 is on the path in any link except the last: true

(A=1 && B=102) Node 1 is on the path in any link except the last, and node 102 is on the path in any link but the first: true

(L=101-102,104-200) Link 101-102, or Link 104-200 is on the path: true

(L=102-101) The path contains link 101-102 or link 102-101: true*

(L=101-102 && L=104-105*) The path contains link (101-102 or 102-101) and link (104-105 or 105-104): true*

(N=101 && N=105-110 && L=101-102) The path contains node 101 and any node in the range of 105-110, and link (101-102 or 102-101): true*

((N=100-105) && (N=8,57 || L=105-104)) The path crosses any node 100-105, and (it crosses node (8 or 57) or link 105-104): false. If L=104-105 instead of L=104-105, the expression would be true.*

Citilabs recommends using nested parentheses to help categorize the comparison sets. With this type of Boolean selection, most any desired combination of path usage can be specified.

Selectlinkgroup also can be used, but would require the user to edit the pathload statements in the "LOAD_TCRPM.INC" script.

LOAD_TCRPM.INC (Cube Script)

```
if(myrun=0)

; ----- General Purpose lanes (Exclude toll lanes, express lanes, HOV ramps, group=4)-----

    PATH=LW.GCOST_DA, peni=1,excludegroup=4,9,VOL[1]=MW[1]

    PATH=LW.GCOST_SR2,peni=1,excludegroup=4,9,VOL[3]=MW[3]

    PATH=LW.GCOST_SR3,peni=1,excludegroup=4,9,VOL[6]=MW[6]

; ----- HOV (exclude toll and HOT - group=3)-----

    ;no SOV

    PATH=LW.GCOST_SR2,peni=1,excludegroup=3,9,VOL[4]=MW[4]

    PATH=LW.GCOST_SR3,peni=1,excludegroup=3,9,VOL[7]=MW[7]

; ----- Pay=either TPK or HOT -----

    PATH=LW.GCOST_DA, peni=1,excludegroup=9, VOL[2]=MW[2]

    PATH=LW.GCOST_SR2,peni=1,excludegroup=9, VOL[5]=MW[5]

    PATH=LW.GCOST_SR3,peni=1,excludegroup=9, VOL[8]=MW[8]

; ----- Trucks and External (EE + EI/IE) -----

    PATH=LW.GCOST_TRK, peni=1,excludegroup=1,9,VOL[9]=mw[9]; trucks, keep out of HOV and HOT

    PATH=LW.GCOST_DA, peni=1,excludegroup=9,VOL[10]=mw[10]; externals (can use toll and HOT)

else

; ----- General Purpose lanes (Exclude toll lanes, express lanes, HOV ramps, group=4)-----

    PATH=LW.GCOST_DA, peni=1,excludegroup=4,9,VOL[1]=MW[1], mw[11]= mw[1], SELECTLINK=(pairs), vol[11]=mw[11]

    PATH=LW.GCOST_SR2,peni=1,excludegroup=4,9,VOL[3]=MW[3], mw[13]= mw[3], SELECTLINK=(pairs), vol[13]=mw[13]

    PATH=LW.GCOST_SR3,peni=1,excludegroup=4,9,VOL[6]=MW[6], mw[16]= mw[6], SELECTLINK=(pairs), vol[16]=mw[16]

; ----- HOV (exclude toll and HOT - group=3)-----

    ;no SOV

    PATH=LW.GCOST_SR2,peni=1,excludegroup=3,9,VOL[4]=MW[4], mw[14]= mw[4], SELECTLINK=(pairs), vol[14]=mw[14]

    PATH=LW.GCOST_SR3,peni=1,excludegroup=3,9,VOL[7]=MW[7], mw[17]= mw[7], SELECTLINK=(pairs), vol[17]=mw[17]

; ----- Pay=either TPK or HOT -----

    PATH=LW.GCOST_DA, peni=1,excludegroup=9, VOL[2]=MW[2], mw[12]= mw[2], SELECTLINK=(pairs), vol[12]=mw[12]

    PATH=LW.GCOST_SR2,peni=1,excludegroup=9, VOL[5]=MW[5], mw[15]= mw[5], SELECTLINK=(pairs), vol[15]=mw[15]

    PATH=LW.GCOST_SR3,peni=1,excludegroup=9, VOL[8]=MW[8], mw[18]= mw[8], SELECTLINK=(pairs), vol[18]=mw[18]

; ----- Trucks and External (EE + EI/IE) -----

    PATH=LW.GCOST_TRK, peni=1,excludegroup=1,9,VOL[9]=mw[9], mw[19]= mw[9], SELECTLINK=(pairs),
vol[19]=mw[19]; trucks, keep out of HOV and HOT

    PATH=LW.GCOST_DA, peni=1,excludegroup=9,VOL[10]=mw[10], mw[20]= mw[10],SELECTLINK=(pairs),
vol[20]=mw[20]; externals (can use toll and HOT)

endif
```

Treasure Coast RPM Highway Load Capacity Restraint and Adjustment (ADJUST PHASE)

Capacity Restraint and Adjustment

The TCRPM4 highway assignment routines are capacity-restrained. Thus, travel times and generalized costs are adjusted between each assignment iteration. Note that the minimum path through the network is always determined by the generalized cost.

The generalized cost is the link time, plus toll facility delays, plus the cost of tolls converted to time using the CTOLL value of time. In TCRPM4, values of CTOLL are calculated in earlier steps for each of the time periods. The model allows for differences in CTOLL values depending on link length and county, but this is not currently used.

The volume/delay function used in the capacity-restraint routine is a modified BPR (from Bureau of Public Roads, the predecessor to FHWA), and the coefficient and exponent values were added to the network in an earlier step, from a lookup table (MVFACTORS.DAT) indexed by facility type.

Volume/delay functions are applied by the LINKCLASS assigned to each network link. LINKCLASS is determined from the link's two-byte facility type (FTC2), and the link's function in the toll road system (plaza, acceleration or deceleration link). LINKCLASS codes are:

- Link Class 1 – Non-regular Toll (it may be a HOT link)
- Link Class 2 - Regular Toll Plaza Link (Never HOT link)
- Link Class 3 - Regular Toll Acceleration Link (Never HOT link)
- Link Class 4 - Regular Toll Deceleration Link (Never HOT link)
- Link Class 5 - Transit Optional Link (Never USED)
- Link Class 6 -Links with junction data; no junction delays added for TCRPM4
- Link Class 9 - Missing FTC2 code links, unlikely to Happen - Just an extra check

TCRPM4 capacities are in units of passenger car equivalents (PCE's). Thus, the volumes used in the volume/delay functions must be PCE's. The following expression calculates PCE's:

$$V = \text{vol}[1] + \text{vol}[2] + \text{vol}[3] + \text{vol}[4] + \text{vol}[5] + \text{vol}[6] + \text{vol}[7] + \text{vol}[8] + \text{vol}[9] * \{\text{PCE}\} + \text{vol}[10]$$

{PCE} is the Cube Key holding the number of passenger cars represented by each heavy truck (1.5).

Capacities are the hourly capacities calculated earlier for each link, multiplied by the effective number of hours in each time period. The number of hours is input as a Cube Key. By time period they are:

- CAPHR_EA = 2.5
- CAPHR_AM = 2.5
- CAPHR_MD = 4.5
- CAPHR_PM = 3

- CAPHR_EV = 6

The BRP equation is applied for link classes 1, 3, 4, and 6. Times are not adjusted for other link classes. The BPR adjustment equation is:

$$\text{Revised time} = (\text{initial free-flow time}) * (1 + \text{bprcoefficient} * (V/C)^{\text{bprexponent}})$$

After each assignment, the generalized costs must be recalculated for the next iteration. These calculations include:

- The calculation of managed lane (HOT) tolls as a function of V/C.
- Calculation of total toll, which is the sum of regular (like Turnpike toll) and HOT toll.
- Calculation of generalized cost, which is the “Revised time” plus toll facility delays, and the cost of tolls converted to time using the CTOLL value of time.

Assignment Model Convergence

The Cube HIGHWAY program combines the loads from each iteration using the Bi-conjugate Frank-Wolfe algorithm. The model proceeds, adjusting times between iteration, until a specified level of convergence is reached. The model is assumed to be converged if more than 6 iterations have been run and,

- The relative gap doesn’t change by at least 0.000001 between iterations or
- The relative gap is below the specified threshold for three consecutive iterations.

The required relative gap for TCRPM4 has been set to 0.0001 for standard model application runs.

(11.1.15) HANET00H.S “Calculate MSA Loads & Times”

VOYAGER NETWORK PROGRAM

This program calculates the “method of successive averages” (MSA) link volumes from the current and past assignments, and then using the volume/delay functions, estimates the resulting travel time on each link. This is done for AM, peak, and off-periods. Note that the MSA volumes are not used as model outputs, but only to calculate the skims for the next iteration.

Input Files

- FILE LINKI[1] = DAILY_LD_tmp.net; MSA loads from last iteration
- FILE LINKI[2] = DAILY_LOADED.net; current assigned volumes
- FILEO NETO = DAILY_MSA.net

Output Files

- FILEO NETO = DAILY_MSA.net

Process Each Link (LINKMERGE) the following logic is applied to each link:

- MERGE RECORD=FALSE; Do not merge records, but combine loads by LINKI index as defined in the code below.

Calculate MSA loads

“@fbi@” is the feedback iteration counter, beginning with one when the model begins. It is incremented by 1, at the end of each model application until successive skims matrices are similar within a specified tolerance. The impact of each successive iteration tends to decline as the value of 1/n gets smaller with each iteration. The new MSA volume is calculated as follows:

	$VMSA_n$	$= VMSA_{n-1} + (1/n) * (CV - VMSA_{n-1})$
Where	n	= current value of @fbi@
	$VMSA_n$	= new MSA volume for the current iteration, n
	$VMSA_{n-1}$	= MSA volume for the previous iteration
	CV	= the model assigned volume for the current iteration

- $AM_V = li.1.AM_V + (1/@fbi@) * (li.2.AM_V - li.1.AM_V);$
 - ; AM_V is the new MSA volume for the current iteration, n
 - ; @fbi@ is current value of @fbi@
 - ; li.2.AM_V is the model assigned volume for the current iteration
 - ; li.1.AM_V is the MSA volume for the previous iteration
 - Off-peak and PM calculations for this pattern
- $PM_V = li.1.PM_V + (1/@fbi@) * (li.2.PM_V - li.1.PM_V)$
- $MD_V = li.1.MD_V + (1/@fbi@) * (li.2.MD_V - li.1.MD_V)$
- $EV_V = li.1.EV_V + (1/@fbi@) * (li.2.EV_V - li.1.EV_V)$
- $EA_V = li.1.EA_V + (1/@fbi@) * (li.2.EA_V - li.1.EA_V)$
- $OF_V = MD_V + EV_V$; define off-peak volume as the sum of midday and evening. Exclude early.

Calculate CAPACITIES for each period using hours in each period as defined in the **Keys** {}.

- $_AMC = li.2.capacity * \{CAPHR_AM\}$

- $_PMC = li.2.capacity * \{CAPHR_PM\}$
- $_MDC = li.2.capacity * \{CAPHR_MD\}$
- $_EVC = li.2.capacity * \{CAPHR_EV\}$
- $_EAC = li.2.capacity * \{CAPHR_EA\}$
- $_OFC = _MDC + _EVC$; define off-peak capacity as the sum of midday and evening. Exclude early.

Set the Initial time as the freeflow time from the current assignment network

- $_t0 = li.2.time$

Set the BPR volume/delay constants. These values were set earlier and are carried in the network.

- $_BPRA = li.2.BPRCOEFFICIENT$
- $_BPRB = li.2.BPREXPONENT$

Set the volume/capacity ratios. V/C ratios are added to the MSA network.

- if($_AMC > 0$)
 - $AM_VCLOSE = AM_V / _AMC$
- ELSE
 - $AM_VCLOSE = 0$
- ENDIF
- IF($_OFC > 0$)
 - $OF_VCLOSE = OF_V / _OFC$
- ELSE
 - $OF_VCLOSE = 0$
- ENDIF
- IF($_PMC > 0$)
 - $PM_VCLOSE = PM_V / _PMC$
- ELSE
 - $PM_VCLOSE = 0$
- ENDIF

Calculate the congested times using the BPR formula. Congested times are added to the MSA network.

- $AM_CONGTIME = _t0 * (1 + _BPRA * (AM_VCLOSE^{_BPRB}))$
- $PM_CONGTIME = _t0 * (1 + _BPRA * (PM_VCLOSE^{_BPRB}))$
- $OF_CONGTIME = _t0 * (1 + _BPRA * (OF_VCLOSE^{_BPRB}))$

Calculate the congested speeds in miles per hour. Congested speeds are added to the MSA network.

- if($AM_CONGTIME > 0$)
 - $AM_CONGSPD = 60 * li.2.distance / AM_CONGTIME$
- ELSE
 - $AM_CONGSPD = 0$
- ENDIF
- if($PM_CONGTIME > 0$)

- $PM_CONGSPD = 60 * li.2.distance / PM_CONGTIME$
- ELSE
 - $PM_CONGSPD = 0$
- ENDIF
- if($OF_CONGTIME > 0$)
 - $OF_CONGSPD = 60 * li.2.distance / OF_CONGTIME$
- ELSE
 - $OF_CONGSPD = 0$
- ENDIF

(11.1.17) HAHWY00F.S 'Congested Skims for feedback

VOYAGER HIGHWAY PROGRAM

The model begins with a set of seed skims save from another model run. This script is run after each iteration is completed, and creates new skim files based on the MSA loads and the volume/delay functions.

Input Files

- FILEI TURNPENI = turns.pen; turn penalties
- FILEI LOOKUPI[1] = HOTTOLL.dbf; HOT lanes toll definitions
- FILEI NETI = DAILY_MSA.net; the network with MSA loads for all 5 periods

Output Files

- MATO[1]= X_AMHSKIMS.mat; AM Period Skims
- MATO[2]= X_OFHSKIMS.mat; Midday Period Skims
- MATO[3]= X_PMHSKIMS.mat; PM Period Skims
 - Each file contains the following matrices, where xx is AM, OF, or PM
 - Non-toll, Drive-alone paths
 - xx_GP_TIME – General purpose congested time
 - xx_GP_FFTIME – General purpose freeflow time
 - xx_GP_DIST – General purpose distance
 - Toll, Drive-alone paths
 - xx_DAT_TIME – Drive alone toll congested time
 - xx_DAT_FFTIME – Drive alone toll freeflow time
 - xx_DAT_DIST – Drive alone toll distance
 - xx_DAT_TOLLCOST – Drive alone toll cost (\$)
 - xx_DAT_TOLLDIST – Drive alone distance on toll links
 - Non-toll HOV, SR2+ paths
 - xx_S2NH_TIME – SR2+ HOV congested time
 - xx_S2NH_FFTIME – SR2+ HOV freeflow time
 - xx_S2NH_DIST – SR2+ HOV distance
 - xx_S2NH_HOVDIST – SR2+ distance on HOV links
 - Toll HOV, SR2+ paths
 - xx_S2TH_TIME – SR2+ Toll HOV Congested time
 - xx_S2TH_FFTIME – SR2+ Toll HOV freeflow time
 - xx_S2TH_DIST – SR2+ Toll HOV distance
 - xx_S2TH_TOLLCOST – SR2+ Toll HOV toll cost (\$)
 - xx_S2TH_TOLLDIST – SR2+ Toll HOV distance on toll links
 - xx_S2TH_HOVDIST – SR2+ Toll HOV distance on HOV links

Define Lookup Tables

- HOT toll rates htrate (LI.HOT,LI.AM_VCLOSE)
 LOOKUPI=1 LIST=N, NAME= htrate,
 LOOKUP[1]= V_C, RESULT= HOT_1, ; curve 1 lookup from V/C return toll rate

;(additional tables can be added for additional HOT systems)
INTERPOLATE =Y; interpolate on the V/C ratio

Process Each Link (LINKMERGE) the following logic is applied to each link:

Set time attributes for toll plazas:

- IF (LI.FTC2==95)
 - LW.FFTIME=0
 - LW.AMTIME=0
 - LW.OFTIME=0;LW.OPTIME=0
 - LW.PMTIME=0
 - LW.SVCTIME=LI.SVCMINUTES+(LI.SVCSECONDS/60)

Not toll plazas

- ELSE
 - LW.FFTIME=LI.FREEFLOWTIME
 - LW.AMTIME=LI.AM_CONGTIME
 - LW.OFTIME=LI.OF_CONGTIME;LW.OPTIME=LI.OF_CONGTIME
 - LW.PMTIME=LI.PM_CONGTIME
 - LW.SVCTIME=0

Set period-specific value of time -- CTOLL is inverse of VOT in min/dollar

- LW.RCTOLLPK=60*(LI.RCTOLL+{DevCtolLPk})
- LW.RCTOLLOP=60*(LI.RCTOLL+{DevCtolLOp})
- IF (LW.RCTOLLPK<0) LW.RCTOLLPK=0.01 ; should not happen
- IF (LW.RCTOLLOP<0) LW.RCTOLLOP=0.01 ; should not happen

Set HOT Link Toll Rates and Express Lanes Tolls

Notes from SERPM7 – but there are no base year TCRPM HOT lanes:

2012 95Express length (one direction) = 10.1 miles

Assume Full Toll for 3+SR SR2 & DA (note- Trucks not permitted), compute weighted HOTTOLL because very few SR3+ are REGISTERED carpools --

- IF (LI.HOT=1-98); All HOT lane systems (none currently in TCRPM)
 - LW.TOLLDIST = LI.DISTANCE
 - AM
 - LW.HOTTOLL_DA_AM = htrate(LI.HOT,LI.AM_VCLOSE); get rates per mile
 - LW.HOTTOLL_SR2_AM = htrate(LI.HOT,LI.AM_VCLOSE)
 - LW.HOTTOLL_SR3_AM = htrate(LI.HOT,LI.AM_VCLOSE)
 - PM
 - LW.HOTTOLL_DA_PM = htrate(LI.HOT,LI.PM_VCLOSE)
 - LW.HOTTOLL_SR2_PM = htrate(LI.HOT,LI.PM_VCLOSE)
 - LW.HOTTOLL_SR3_PM = htrate(LI.HOT,LI.PM_VCLOSE)
 - OFF PEAK
 - LW.HOTTOLL_DA_OP = htrate(LI.HOT,LI.OF_VCLOSE)
 - LW.HOTTOLL_SR2_OP = htrate(LI.HOT,LI.OF_VCLOSE)
 - LW.HOTTOLL_SR3_OP = htrate(LI.HOT,LI.OF_VCLOSE)

- ELSE; (not HOT)
 - LW.HOTTOLL_DA_AM=0
 - LW.HOTTOLL_SR2_AM=0
 - LW.HOTTOLL_SR3_AM=0
 - LW.HOTTOLL_DA_OP =0
 - LW.HOTTOLL_SR2_OP=0
 - LW.HOTTOLL_SR3_OP=0
 - LW.HOTTOLL_DA_PM =0
 - LW.HOTTOLL_SR2_PM=0
 - LW.HOTTOLL_SR3_PM=0
- ENDIF

Set total toll cost (\$)

- LW.TOLL_AMDA = LI.CARTOLL + LW.HOTTOLL_DA_AM * LI.DISTANCE
- LW.TOLL_AMSR2 = LI.CARTOLL + LW.HOTTOLL_SR2_AM * LI.DISTANCE
- LW.TOLL_AMSR3 = LI.CARTOLL + LW.HOTTOLL_SR3_AM * LI.DISTANCE
- LW.TOLL_PMDA = LI.CARTOLL + LW.HOTTOLL_DA_PM * LI.DISTANCE
- LW.TOLL_PMSR2 = LI.CARTOLL + LW.HOTTOLL_SR2_PM * LI.DISTANCE
- LW.TOLL_PMSR3 = LI.CARTOLL + LW.HOTTOLL_SR3_PM * LI.DISTANCE
- LW.TOLL_OPDA = LI.CARTOLL + LW.HOTTOLL_DA_OP * LI.DISTANCE
- LW.TOLL_OPSR2 = LI.CARTOLL + LW.HOTTOLL_SR2_OP * LI.DISTANCE
- LW.TOLL_OPSR3 = LI.CARTOLL + LW.HOTTOLL_SR3_OP * LI.DISTANCE

Set toll and HOV distance for each vehicle class

Express lanes are toll lanes for drive alone and shared ride 3+

- IF(LI.FTC1=80) ; HOV Lanes
 - LW.HOV_DIST = LI.DISTANCE
- ENDIF
- IF(LI.IsToll>0 | LI.HOT>0); IsToll means you cannot drive on this link without paying a toll
 - LW.TOLLDIST = LI.DISTANCE
- ENDIF

Set mode-specific toll distances

- LW.TOLLDIST_DA = LW.TOLLDIST
- LW.TOLLDIST_SR2 = LW.TOLLDIST
- LW.TOLLDIST_SR3 = LW.TOLLDIST
- LW.HOVDIST_SR2 = LW.HOV_DIST
- LW.HOVDIST_SR3 = LW.HOV_DIST

Set impedance to the sum of time (congested link time and any toll service time) and the time-equivalent of toll cost. Used to determine paths.

- LW.GENCOST_AMDA = LW.AMTIME + LW.SVCTIME + LW.TOLL_AMDA * LW.RCTOLLPK

- $LW.GENCOST_AMSR2 = LW.AMTIME + LW.SVCTIME + LW.TOLL_AMSR2 * LW.RCTOLLPK$
- $LW.GENCOST_AMSR3 = LW.AMTIME + LW.SVCTIME + LW.TOLL_AMSR3 * LW.RCTOLLPK$
- $LW.GENCOST_PMDA = LW.PMETIME + LW.SVCTIME + LW.TOLL_PMDA * LW.RCTOLLPK$
- $LW.GENCOST_PMSR2 = LW.PMETIME + LW.SVCTIME + LW.TOLL_PMSR2 * LW.RCTOLLPK$
- $LW.GENCOST_PMSR3 = LW.PMETIME + LW.SVCTIME + LW.TOLL_PMSR3 * LW.RCTOLLPK$
- $LW.GENCOST_OPDA = LW.OFTIME + LW.SVCTIME + LW.TOLL_OPDA * LW.RCTOLLOP$
- $LW.GENCOST_OPSR2 = LW.OFTIME + LW.SVCTIME + LW.TOLL_OPSR2 * LW.RCTOLLOP$
- $LW.GENCOST_OPSR3 = LW.OFTIME + LW.SVCTIME + LW.TOLL_OPSR3 * LW.RCTOLLOP$

SET LINK EXCLUSION GROUPS BY MODE AND TIME PERIOD, unlike assignment, each time period must be allocated to a different set of codes.

(note that each link can be assigned to up to 32 groups)

AM DA NO-TOLL, exclude toll lanes, express lanes, AM HOV ramps

- IF(LI.HOT>0 | LI.FTC2=83 | LI.FTC2=84); HOT lanes and HOV access ramps
 - addtogroup=1
- ENDIF
- IF(LI.FTC1=90 & LI.ISRCTOLLLINK=0); Turnpike links
 - addtogroup=1
- ENDIF

AM DA TOLL, exclude AM HOV ramps

- IF(LI.FTC2=83 | LI.FTC2=84); AM HOV ramps
 - addtogroup=2
- ENDIF

SR2+ HOV NO-TOLL, exclude toll lanes, express lanes

- IF(LI.HOT>0); HOT lanes
 - addtogroup=3
- ENDIF
- IF(LI.FTC1=90 & LI.ISRCTOLLLINK=0); Turnpike links
 - addtogroup=3
- ENDIF

OFF-PEAK DA NO-TOLL, exclude toll lanes, express lanes, HOV ramps

- IF(LI.HOT>0 | LI.FTC2=86); HOT lanes and HOV access ramps
 - addtogroup=4
- ENDIF
- IF(LI.FTC1=90 & LI.ISRCTOLLLINK=0) ; Turnpike links
 - addtogroup=4
- ENDIF

OFF-PEAK DA TOLL, exclude HOV ramps

- IF(LI.FTC2=86); HOV access ramps
 - addtogroup=5
- ENDIF

PM DA NO-TOLL, exclude toll lanes, express lanes, PM HOV ramps

- IF(LI.HOT>0 | LI.FTC2=83 | LI.FTC2=85) ; HOT lanes and HOV access ramps
 - addtogroup=6
- ENDIF
- IF(LI.FTC1=90 & LI.ISRCTOLLINK=0) ; Turnpike links
 - addtogroup=6
- ENDIF
- PM DA TOL; exclude PM HOV ramps
- IF(LI.FTC2=83 | LI.FTC2=85); HOV access ramps
 - addtogroup=7
- ENDIF

OFF-PEAK direction REVERSIBLE lanes shutdown; Transit non-Highway opt links(59,69); Dummy HOT ramps

- IF (((LI.FTC2=49,59,69) || (LI.REVERSIBLE=1,2) || (LI.HOT=99)) & LI.ISREQTLINKSKIM=0)
ADDTOGROUP=9 ; offpeak

AM & PM direction REVERSIBLE lanes shutdown; Transit non-Highway opt links(59,69); Dummy HOT ramps

- IF (((LI.FTC2=49,59,69) || (LI.REVERSIBLE=2) || (LI.HOT=99)) & LI.ISREQTLINKSKIM=0)
ADDTOGROUP=31 ; AM

AM&PM direction REVERSIBLE lanes shutdown; Transit non-Highway opt links(59,69); Dummy HOT ramps

- IF (((LI.FTC2=49,59,69) || (LI.REVERSIBLE=1) || (LI.HOT=99)) & LI.ISREQTLINKSKIM=0)
ADDTOGROUP=32 ; PM

Process each trip interchange (ILOOP)

COMPUTE INTERZONAL SKIMS – note that generalized cost is always used for the minimum path

AM SKIMS

Non-toll, Drive-alone

- PATH=LW.GENCOST_AMDA,peni=1,excludegroup=1,31,
- MW[01]=PATHTRACE(LW.AMTIME,1),NOACCESS=999.99,dec=2, ;TIME Skim includes pen set 1
- MW[02]=PATHTRACE(LW.FFTIME,1),NOACCESS=999.99,dec=2, ;TIME Skim includes pen set 1
- MW[03]=PATHTRACE(LI.DISTANCE),NOACCESS=999.99,dec=2

Toll, Drive-alone

- PATH=LW.GENCOST_AMDA,peni=1,excludegroup=2,31,
- MW[04]=PATHTRACE(LW.AMTIME,1),NOACCESS=999.99,dec=2, ;TIME Skim includes pen set 1
- MW[05]=PATHTRACE(LW.FFTIME,1),NOACCESS=999.99,dec=2, ;TIME Skim includes pen set 1

- MW [06]=PATHTRACE(LI.DISTANCE),NOACCESS=999.99,dec=2,
- MW [07]=PATHTRACE(LW.TOLL_AMDA),NOACCESS=999.99,dec=2,
- MW[08]=PATHTRACE(LW.TOLLDIST_DA),NOACCESS=999.99,dec=2

Non-toll HOV, SR2+

- PATH=LW.GENCOST_AMSR2,peni=1,excludegroup=3,31,
- MW[09]=PATHTRACE(LW.AMTIME,1),NOACCESS=999.99,dec=2, ;TIME Skim includes pen set 1
- MW[10]=PATHTRACE(LW.FFTIME,1),NOACCESS=999.99,dec=2, ;TIME Skim includes pen set 1
- MW[11]=PATHTRACE(LI.DISTANCE),NOACCESS=999.99,dec=2,
- MW[12]=PATHTRACE(LW.HOVDIST_SR2),NOACCESS=999.99,dec=2

Toll, HOV, SR2+

- PATH=LW.GENCOST_AMSR2,peni=1,excludegroup=31,
- MW[13]=PATHTRACE(LW.AMTIME,1),NOACCESS=999.99,dec=2, ;TIME Skim includes pen set 1
- MW[14]=PATHTRACE(LW.FFTIME,1),NOACCESS=999.99,dec=2, ;TIME Skim includes pen set 1
- MW[15]=PATHTRACE(LI.DISTANCE),NOACCESS=999.99,dec=2,
- MW[16]=PATHTRACE(LW.TOLL_AMSR2),NOACCESS=999.99,dec=2,
- MW[17]=PATHTRACE(LW.TOLLDIST_SR2),NOACCESS=999.99,dec=2,
- MW[18]=PATHTRACE(LW.HOVDIST_SR2),NOACCESS=999.99,dec=2

Off-Peak SKIMS

Non-toll, Drive-alone

- PATH=LW.GENCOST_OPDA,peni=1,excludegroup=4,9,
- MW[21]=PATHTRACE(LW.OFTIME,1),NOACCESS=999.99,dec=2, ;TIME Skim includes pen set 1
- MW[22]=PATHTRACE(LW.FFTIME,1),NOACCESS=999.99,dec=2, ;TIME Skim includes pen set 1
- MW[23]=PATHTRACE(LI.DISTANCE),NOACCESS=999.99,dec=2

Toll, Drive-alone

- PATH=LW.GENCOST_OPDA,peni=1,excludegroup=5,9,
- MW[24]=PATHTRACE(LW.OFTIME,1),NOACCESS=999.99,dec=2, ;TIME Skim includes pen set 1
- MW[25]=PATHTRACE(LW.FFTIME,1),NOACCESS=999.99,dec=2, ;TIME Skim includes pen set 1
- MW [26]=PATHTRACE(LI.DISTANCE),NOACCESS=999.99,dec=2,
- MW [27]=PATHTRACE(LW.TOLL_OPDA),NOACCESS=999.99,dec=2,
- MW[28]=PATHTRACE(LW.TOLLDIST_DA),NOACCESS=999.99,dec=2

Non-toll HOV, SR2+

- PATH=LW.GENCOST_OPSR2,peni=1,excludegroup=3,9,
- MW[29]=PATHTRACE(LW.OFTIME,1),NOACCESS=999.99,dec=2, ;TIME Skim includes pen set 1
- MW[30]=PATHTRACE(LW.FFTIME,1),NOACCESS=999.99,dec=2, ;TIME Skim includes pen set 1
- MW[31]=PATHTRACE(LI.DISTANCE),NOACCESS=999.99,dec=2,
- MW[32]=PATHTRACE(LW.HOVDIST_SR2),NOACCESS=999.99,dec=2

Toll HOV, SR2+

- PATH=LW.GENCOST_OPSR2,peni=1,excludegroup=9,
- MW[33]=PATHTRACE(LW.OFTIME,1),NOACCESS=999.99,dec=2, ;TIME Skim includes pen set 1
- MW[34]=PATHTRACE(LW.FFTIME,1),NOACCESS=999.99,dec=2, ;TIME Skim includes pen set 1
- MW[35]=PATHTRACE(LI.DISTANCE),NOACCESS=999.99,dec=2,
- MW[36]=PATHTRACE(LW.TOLL_OPSR2),NOACCESS=999.99,dec=2,
- MW[37]=PATHTRACE(LW.TOLLDIST_SR2),NOACCESS=999.99,dec=2,
- MW[38]=PATHTRACE(LW.HOVDIST_SR2),NOACCESS=999.99,dec=2

PM SKIMS

Non-toll, Drive-alone

- PATH=LW.GENCOST_PMDA,peni=1,excludegroup=6,32,
- MW[41]=PATHTRACE(LW.PMTIME,1),NOACCESS=999.99,dec=2, ;TIME Skim includes pen set 1
- MW[42]=PATHTRACE(LW.FFTIME,1),NOACCESS=999.99,dec=2, ;TIME Skim includes pen set 1
- MW[43]=PATHTRACE(LI.DISTANCE),NOACCESS=999.99,dec=2

Toll, Drive-alone

- PATH=LW.GENCOST_PMDA,peni=1,excludegroup=7,32,
- MW[44]=PATHTRACE(LW.PMTIME,1),NOACCESS=999.99,dec=2, ;TIME Skim includes pen set 1
- MW[45]=PATHTRACE(LW.FFTIME,1),NOACCESS=999.99,dec=2, ;TIME Skim includes pen set 1
- MW[46]=PATHTRACE(LI.DISTANCE),NOACCESS=999.99,dec=2,
- MW[47]=PATHTRACE(LW.TOLL_PMDA),NOACCESS=999.99,dec=2,
- MW[48]=PATHTRACE(LW.TOLLDIST_DA),NOACCESS=999.99,dec=2

Non-toll HOV, SR2+

- PATH=LW.GENCOST_PMSR2,peni=1,excludegroup=3,32,
- MW[49]=PATHTRACE(LW.PMTIME,1),NOACCESS=999.99,dec=2, ;TIME Skim includes pen set 1
- MW[50]=PATHTRACE(LW.FFTIME,1),NOACCESS=999.99,dec=2, ;TIME Skim includes pen set 1
- MW[51]=PATHTRACE(LI.DISTANCE),NOACCESS=999.99,dec=2,
- MW[52]=PATHTRACE(LW.HOVDIST_SR2),NOACCESS=999.99,dec=2

; ----- Toll HOV, SR2+ -----

- PATH=LW.GENCOST_PMSR2,peni=1,excludegroup=32,
- MW[53]=PATHTRACE(LW.PMTIME,1),NOACCESS=999.99,dec=2, ;TIME Skim includes pen set 1
- MW[54]=PATHTRACE(LW.FFTIME,1),NOACCESS=999.99,dec=2, ;TIME Skim includes pen set 1
- MW[55]=PATHTRACE(LI.DISTANCE),NOACCESS=999.99,dec=2,
- MW[56]=PATHTRACE(LW.TOLL_PMSR2),NOACCESS=999.99,dec=2,
- MW[57]=PATHTRACE(LW.TOLLDIST_SR2),NOACCESS=999.99,dec=2,
- MW[58]=PATHTRACE(LW.HOVDIST_SR2),NOACCESS=999.99,dec=2

COMPUTE INTERZONAL SKIMS - Average of the TWO Nearest Zones

AM SKIMS

- MW[01][i] = lowest(01,2) * 0.25 ;Intrazonal AM DA Time
- MW[02][i] = lowest(02,2) * 0.25 ;Intrazonal AM DA FFTime

- $MW[03][i] = \text{lowest}(03,2) * 0.25$;Intrazonal AM DA Distance
- $MW[07][i] = 999.0$;No IZ TOLL

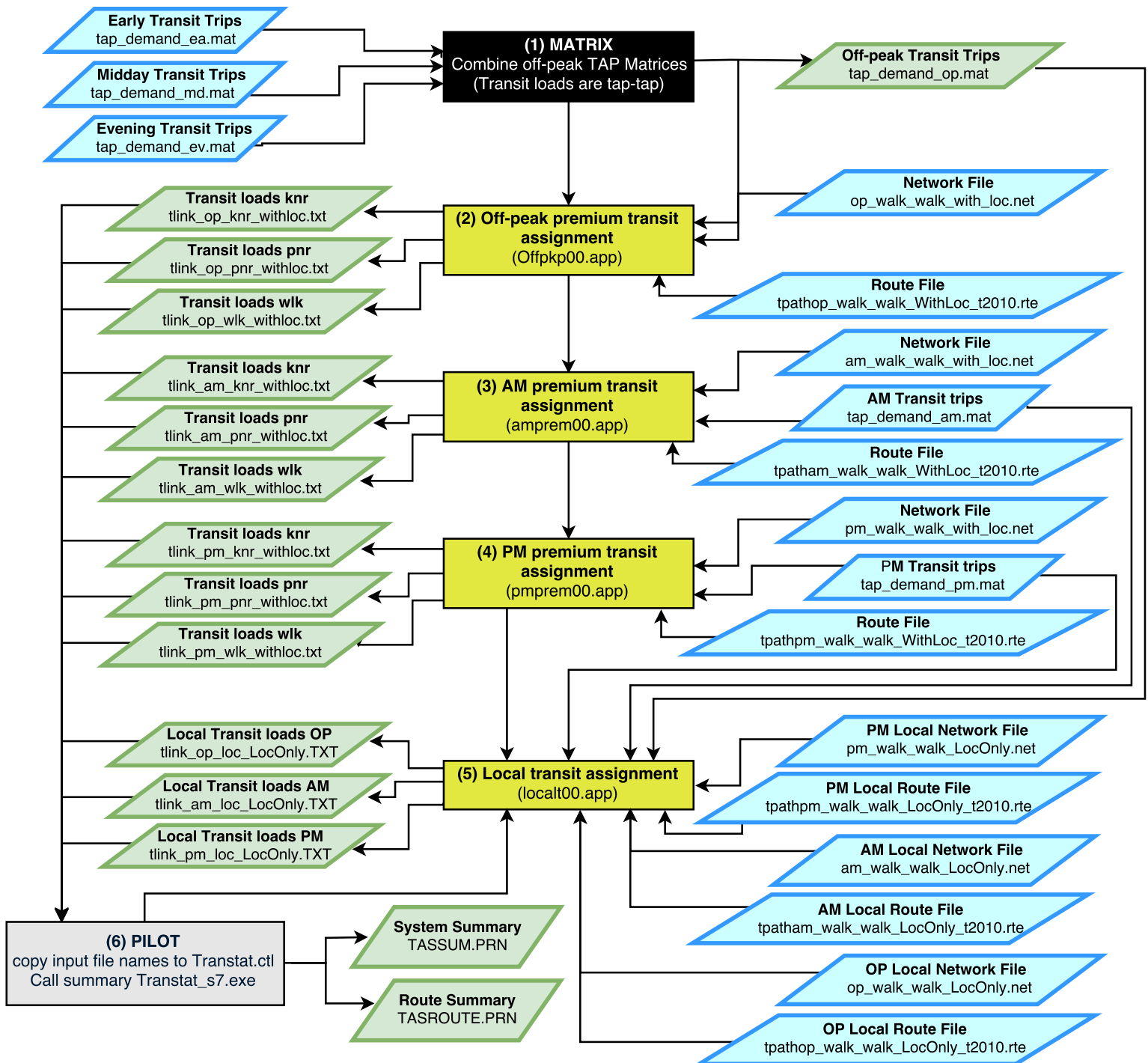
Off-Peak Skims

- $MW[21][i] = \text{lowest}(21,2) * 0.25$;Intrazonal Off-peak DA Time
- $MW[22][i] = \text{lowest}(22,2) * 0.25$;Intrazonal Off-peak DA FFTime
- $MW[23][i] = \text{lowest}(23,2) * 0.25$;Intrazonal Off-peak DA Distance
- $MW[27][i] = 999.0$;No IZ TOLL

PM Skims

- $MW[41][i] = \text{lowest}(41,2) * 0.25$;Intrazonal PM DA Time
- $MW[42][i] = \text{lowest}(42,2) * 0.25$;Intrazonal PM DA FFTime
- $MW[43][i] = \text{lowest}(43,2) * 0.25$;Intrazonal PM DA Distance
- $MW[47][i] = 999.0$;No IZ TOLL

(14) Transit Assignment



This application combines early, midday, and evening transit trip tables to create an off-peak transit trip table. AM, PM, and off-peak trip tables, for local and premium services are loaded onto the appropriate networks. For premium services, loads are developed for drop-off (KnR), park and ride (PnR), and walk access paths by time of day. For local services, loads are developed by time of day on walk access paths. The transit loads are then fed into the TranStat program, and transit summaries are produced by transit system, and by transit route.