Progression (version 4.0.6)

Revision History
14 June 2011 Created by Volpe Center

The Progression program calculates offsets for a set of fixed time traffic signals. Progression can be used to

1. Set phase offsets for fixed timed signals in progression groups.
2. Create progression groups based on thru movements between fixed timed signals.
3. Use network travel times or user defined progression speed to calculate the offsets.
4. Set different phase offsets for different time periods.
5. Optimize based on percent thru, network speed, or vehicle hours of travel.
6. Limit offset calculations to progression groups defined in a link equivalence file.
7. Give priority to specific progression groups or fix the processing order.
8. Create an ArcView shape file with performance measures for each progression group.

Progression is a console-based program that runs in a command window on either Windows or Linux. The command syntax is:

```
Progression [-flag] [control_file]
```

The control_file is the file name of an ASCII file that contains the control strings expected by the program. The control_file is optional. If a file name is not provided, the program will prompt the user to enter a file name. The flag parameters are also optional. Any combination of the following flag parameters can be included on the command line:

-Q[uiet] = execute without screen messages
-H[elp] = show program syntax and control keys
-K[eyCheck] = list unrecognized control file keys
-P[ause] = pause before exiting
-N[oPause] = never pause before exiting
-B[atch] = execute in batch processing mode

The program automatically creates a printout file based on the control_file name. If the file name includes an extension, the extension is removed and ".prn" is added. The printout file will be created in the current working directory and will overwrite an existing file with the same name.

Known Gaps in this Document

Example is only provided for an extremely simple network.

Control File Examples

Control parameters are defined using a control key followed by a string or number. The control parameters can be specified in any order. If a given key is defined more than once, the last instance of the key is used. Note that comment lines or extraneous keys can be included in the file. They will be ignored by the program.

EXAMPLE 1 SET OFFSETS ON FIXED TIME SIGNALS

```
    TITLE        Signal Progression
    PROJECT_DIRECTORY

    DEFAULT_FILE_FORMAT        TAB_DELIMITED

#---- Input Files ----

    NET_DIRECTORY        ThreeRoadNetwork
    NET_NODE_TABLE        Node.txt
    NET_LINK_TABLE        Link.txt
    NET_LANE_CONNECTIVITY_TABLE   Lane_Connectivity.txt
    NET_SIGNALIZED_NODE_TABLE        Signalized_Node.txt
    NET_PHASING_PLAN_TABLE        Phasing_Plan.txt
    NET_TIMING_PLAN_TABLE        Timing_Plan.txt

    #NET_SHAPE_TABLE      Input_Shape.txt
    NET_UNSIGNALIZED_NODE_TABLE        Unsignalized_Node.txt

    LINK_EQUIVALENCE_FILELinkEquiv.txt
    GROUP_PERIOD_WEIGHT_FILE     GroupWeight.txt

#---- Output Files ----

    NEW_SIGNALIZED_NODE_TABLE     New_Signalized_NodeThreeRoad.txt
    ARCVIEW_PROGRESSION_FILE     NewProgression.shp

#---- Parameters ----
   CLEAR_EXISTING_OFFSETS   TRUE
   #LINK_DELAY_FILE    ThreeRoadNetwork/Link_Delay.txt
   PROGRESSION_TIME_PERIODS   0:00..12:00
   PROGRESSION_PERIOD_SPEED  12

   PROGRESSION_REPORT_1  GROUP_PERIOD_WEIGHTS
   PROGRESSION_REPORT_2  PRINT_LINK_EQUIVALENCIES
```

Control File Parameters

The keys recognized by the **Progression** program are listed below. These keys can be defined in a variety of different ways to perform different tasks.

Required Keys

NET_NODE_TABLE

The node table specifies the name of the TRANSIMS node file within the network directory. The full path and file name for the node table is constructed by appending the value of this key to the value of the NET_DIRECTORY key.

NET_LINK_TABLE

The link table key specifies the name of the TRANSIMS link file within the network directory. The full path and file name for the link table is constructed by appending the value of this key to the value of the NET_DIRECTORY key. The links used in Example 1 are shown below. Figure 1 (on page 2) illustrates this example network.

LINK	STREET	ANODE	BNODE	LENGTH	LANES_AB	LANES_BA	USE
2	Main	20	26	750	1	1	WALK/AUTO/TRUCK/BUS
3	Main	26	27	300	1	1	WALK/AUTO/TRUCK/BUS
4	Main	27	21	750	1	1	WALK/AUTO/TRUCK/BUS

NET_LANE_CONNECTIVITY_TABLE

The network lane connectivity table key is required. It specifies the name of the TRANSIMS lane connectivity file within the network directory. The full path and file name for the lane connectivity table is constructed by appending the value of this key to the value of the NET_DIRECTORY key.

NET_SIGNALIZED_NODE_TABLE

The network signalized node table key is required. It specifies the name of the TRANSIMS signalized node file within the network directory. The full path and file name for the signalized node table is constructed by appending the value of this key to the value of the NET_DIRECTORY key. The signalized nodes used in Example 1 are shown below. They are both fixed time signals of type T.

NODE	START	TIMING	TYPE	RINGS	OFFSET	COORDINATOR	NOTES
26	0:00	1	T	S	0	1	4 Phase Timed
27	0:00	2	T	S	0	2	4 Phase Timed

NET_TIMING_PLAN_TABLE

The network timing plan table key is required. It specifies the name of the TRANSIMS timing plan file within the network directory. The full path and file name for the timing plan table is constructed by appending the value of this key to the value of the NET_DIRECTORY key. The timing plan used in Example 1 is shown below. Note that the cycle lengths for the two signals are the same.

TIMING	PHASE	NEXT_ PHASE	MIN_ GREEN	MAX_ GREEN	EXT_ GREEN	YELLOW	RED_ CLEAR	RING	BARRIER	NOTES
1	1	2	12	0	0	0	0	1	0	NODE 26
1	2	3	24	0	0	3	1	0	0	NODE 26
1	3	4	12	0	0	0	0	0	0	NODE 26
1	4	1	40	0	0	3	1	0	0	NODE 26
2	1	2	12	0	0	0	0	1	0	NODE 27

2	2	3	24	0	0	3	1	0	0	NODE 27
2	3	4	12	0	0	0	0	0	0	NODE 27
2	4	1	40	0	0	3	1	0	0	NODE 27

NET_PHASING_PLAN_TABLE

The network phasing plan table key is required. It specifies the name of the TRANSIMS phasing plan file within the network directory. The full path and file name for the phasing plan table is constructed by appending the value of this key to the value of the NET_DIRECTORY key. The relevant portion (thru east-west movements) of the Example 1 Phasing_Plan is shown below.

NODE	TIMING	PHASE	IN_LINK	OUT_LINK	PROTECTION	DETECTORS	NOTES
26	1	4	3	2	P	0	Protected Thru
26	1	4	2	3	P	0	Protected Thru
27	2	4	4	3	P	0	Protected Thru
27	2	4	3	4	P	0	Protected Thru

NEW_SIGNALIZED_NODE_TABLE

The network signalized node table key is required. It specifies the name of the new output TRANSIMS signalized node file. The full path and file name for the phasing plan table is constructed by appending the value of this key to the value of the NET_DIRECTORY key. This file contains the new offsets. In the example below, the second node is offset by 25 seconds, which is what one would expect with a primary eastbound flow, a distance between nodes of 300 m, and a speed of 12 m / s.

NODE	START	TIMING	TYPE	RINGS	OFFSET	COORDINATOR	NOTES
26	0:00	1	T	S	0	1	4 Phase Timed
27	0:00	2	T	S	25	2	4 Phase Timed

Optional Keys

TITLE

Any text string can be used on this line. This text is printed on the top of each output page.

REPORT_FILE

The report file name is optional. If a file name is not provided, the program automatically creates a report file name based on the input control file name plus the partition number. The report file will overwrite an existing file with the same name if the Report Flag key is False or not specified.

REPORT_FLAG

The report flag key is optional. If it is specified as Yes or True, the report file or default printout file will be opened in 'Append' mode rather than 'Create' mode. This permits the user to consolidate the output of several programs into a single report file.

MAX_WARNING_MESSAGES

When the program generates a warning message, a counter is incremented and the total number of warning messages is reported and a warning return coded (2) is set at the end of the execution. By

default the program prints up to 100,000 warning messages to the print-out file. If more than 100,000 warning messages are sent, the program stops printing additional messages to the file or terminates the program with an error message.

MAX_WARNING_EXIT_FLAG

If the maximum number of warning messages is exceeded, this flag directs the program in what to do. If the flag is TRUE (the default), the program is terminated with an error message about the warning messages. If the flag is FALSE, the program continues execution, but no additional warning messages are sent to the screen or written to the printout file. The warning message counter continues to count the messages and reports the total at the end of the execution.

PROJECT_DIRECTORY

The project directory key is not required. If it is specified, it is added to all non-network file names required by the program. If it is not specified, all non-network file names should fully specify the file path.

DEFAULT_FILE_FORMAT

Default format for files other than network files. Default is VERSION3. Other possible values include BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, LANL and SQLITE3.

NET_DIRECTORY

The network directory key is not required. If it is specified, it is added to all network table names. If it is not specified, the network table names should fully specify the file path.

NET_SHAPE_TABLE

The shape table key is optional. If specifies the name of the TRANSIMS shape file within the network directory. The full path and file name for the shape table is constructed by appending the value of this key to the value of the NET_DIRECTORY key. If the shape table is provided, all links and link related offsets are drawn based on the shape of the link.

NET_UNSIGNALIZED_NODE_TABLE

The unsignalized node table key is optional. It specifies the name of the TRANSIMS unsignalized node file within the network directory. The full path and file name for the unsignalized node table is constructed by appending the value of this key to the value of the NET_DIRECTORY key.

NEW_DIRECTORY

The new directory key is not required. If it is specified, it is added to all output table names. If it is not specified, the output table names should fully specify the file path.

LINK_DELAY_FILE

The link delay file key is optional. If the key is provided, the program uses the volume information in the link delay file.

LINK_EQUIVALENCE_FILE

The new link equivalence file key is appended to the PROJECT_DIRECTORY key to specify the file name for the link equivalence file used by the program. It is used to set up groups of links (for example, progression groups). A sample file, along with a sketch of the network, is shown below. There are two groups: Group 1 represents eastbound flow, while Group 2 represents westbound flow.

```
1 0 Eastbound
1 1 2, 3, 4
2 0 Westbound
2 1 -4, -3, -2
```

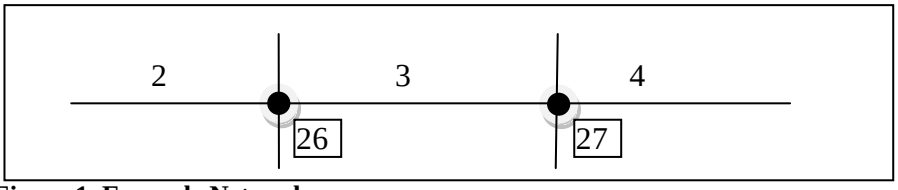

Figure 1 Example Network

CLEAR_EXISTING_OFFSETS

True/False {true/false/yes/no/1/0} variable with a default of FALSE. Determines if existing offsets should be cleared.

EVALUATE_EXISTING_OFFSETS

True/False {true/false/yes/no/1/0} variable with a default of FALSE. Determines if existing offsets should be evaluated.

PROGRESSION_TIME_PERIODS

One or more time ranges for progression, with a default of "All". Time ranges are expressed as 0:00..6:00, 16:00..18:00, etc.

PROGRESSION_PERIOD_SPEED

A set of speeds to be used in the progression calculations. These fields accept a comma separated list of values (e.g., 0, 10, 0, 10) to define the progression speed value for each time period. A zero indicates that the travel times in the link delay file or free flow travel times will be used for the phase offset calculations. The example above uses the link delay times for periods 1 and 3 and 10 meters per second for all other periods. The last in the string is used for all subsequent time periods.

OPTIMIZATION_METHOD

Options include {PERCENT_THRU, NETWORK_SPEED, VEHICLE_HOURS}, with a default of PERCENT_THRU. By default, the process iterates through starting from each progression group and cascading through all of the other progression groups that intersect that group. A composite performance measure is calculated for each sequence and the sequence with the best performance measure is selected for output. This key defines the method of calculating the performance measure. The algorithm will attempt to maximize the percent thru bandwidth or the overall travel speed of the network, or minimize the vehicle hours of travel based on link volumes included in the link delay file.

GROUP_PERIOD_WEIGHT_FILE

A file that provides weighting factors by time period for each link group. An example appears below, with groups 1 and 2. Group 1 is weighted at 60% and Group 2 at 40%.

```
1       60
2       40
```

KEEP_LINK_GROUP_ORDER

True/ False variable {true/false/yes/no/1/0} with a default of FALSE. Manually overrides the order of the progression settings based on the record order in the link equivalence file. If Keep_Link_Group_Order is TRUE, a link equivalence file is required.

ARCVIEW_PROGRESSION_FILE

Filename.shp. The name of an output arcview shape file that contains progression information. The file includes vectors with fields of time period, progression order, group, percent, travel time, length, speed and vehicle hours traveled. (control.cpp)

LINK_DIRECTION_OFFSET

Defaults to 5 meters. Range is 0 to 15 meters.

NET_DEFAULT_FORMAT

Default format for network files. The default file format is set by DEFAULT_FILE_FORMAT. Other options include VERSION3, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, LANL.

NET_*_FORMAT

The file format key enables the user to specify the input format for an input network file. Replace the * with any of the network file types: node, link, activity_location, lane_use. The default file format is set by NET_DEFAULT_FORMAT. Other options include VERSION3, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, LANL.

LINK_DELAY_FORMAT

Defaults to the default file format.

NEW_DEFAULT_FORMAT

Default format for new output files. The default file format is set by DEFAULT_FILE_FORMAT. Other options include VERSION3, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, LANL.

NEW_SIGNALIZED_NODE_FORMAT

The file format key enables the user to specify the format for output files. Replace the * with any of the output file types: link_trip_end, location_trip_end, zone_trip_end, trip_table. The default file format is set by NEW_DEFAULT_FORMAT. The format options include VERSION3, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, LANL.

PROGRESSION_REPORT_#

Two reports are available:
PRINT_LINK_EQUIVALENCIES
GROUP_PERIOD_WEIGHTS

Sample Printouts

Sample printout files generated by the Progression program are shown below. Each printout is an ASCII text file with a maximum of 95 characters per line and 65 lines per page. The file can be viewed or printed using a variety of text editors. For best results in a word processor, use a 10-point Courier font and 0.5 inch margins on all sides. In the examples, headings for reports have been reformatted to improve readability.

Example 1

```
*****************************************
|                                       |
|       Progression - Version 4.0.6     |
|   Copyright (c) 2009 by AECOM Consult |
|         Tue Jun 21 15:12:27 2011      |
|                                       |
*****************************************

Control File = Progression.ctl
Report_File  = Progression.prn (Create)

Signal Progression

Default File Format = TAB_DELIMITED

Network Directory = ThreeRoadNetwork
Node File = ThreeRoadNetwork\Node.txt
Link File = ThreeRoadNetwork\Link.txt
Lane Connectivity File = ThreeRoadNetwork\Lane_Connectivity.txt
Unsignalized Node File = ThreeRoadNetwork\Unsignalized_Node.txt
Signalized Node File = ThreeRoadNetwork\Signalized_Node.txt
Phasing Plan File = ThreeRoadNetwork\Phasing_Plan.txt
Timing Plan File = ThreeRoadNetwork\Timing_Plan.txt

New Signalized Node File = New_Signalized_NodeThreeRoad.txt

Link Equivalence File = LinkEquiv.txt

Clear Existing Offsets = TRUE

Progression Time Periods = 0:00..12:00

Progression Period Speed = 12.0 meters per second

Optimization Method = PERCENT_THRU

Group Period Weight File = GroupWeight.txt

New ArcView Progression File = NewProgression.shp
Link Direction Offset = 5.00 meters

Progression Reports:  1. GROUP_PERIOD_WEIGHTS
                      2. PRINT_LINK_EQUIVALENCIES

Link Equivalence
[Eastbound               ]   1 = 2, 3, 4
[Westbound               ]   2 = -4, -3, -2
```

```
Number of Node File Records = 14

Number of Link File Records = 13
Number of Directional Links = 26

Number of Lane Connectivity File Records = 36

Number of Unsignalized Node File Records = 18

Number of Timing Plan File Records = 8

Number of Signalized Node File Records = 2

Number of Phasing Plan File Records = 40

Number of Group Period Weight Records = 2

Group Period Weight Report

Group  Period 1
    1     60.00
    2     40.00

**** Period 1, Group 1 [Eastbound] Weight 60.00 ****
Node=      26, Arrival=  0,  SET, Departure=  0, Intersect= 2
Node=      27, Arrival= 25,  SET, Departure= 25, Intersect= 2

**** Period 1, Group 2 [Westbound] Weight 40.00 ****
Node=      27, Arrival= 25, THRU, Departure= 25, Intersect= 1
Node=      26, Arrival= 50, STOP, Departure=  0, Intersect= 1

Period 1 Performance Measure = 80000

Number of Fixed Time Signals = 2
Number of Updated Offsets = 2
Number of Signal Groups = 2
Number of Time Periods = 1
Non-Stop Thru Movements = 75.0%

Tue Jun 21 15:12:27 2011 -- Process Complete (0:00:00)
```