FAST-TrIPs User's Manual

Tyler Beduhn, Alireza Khani
Network Modeling Center, University of Texas at Austin
Nov 2014

1. Introduction

FAST-TrIPs (Flexible Assignment and Simulation Tool for Transit and Intermodal Passengers) is a dynamic transit assignment model for regional planning analysis. It assigns each transit traveler a specific transit route considering the published vehicle schedule. It also has the capability of taking into account the congestion and simulated vehicle trajectories through integration with a dynamic traffic assignment (DTA) model. The model was developed as a part of an integrated advanced transportation model in SHRP2 C10B project sponsored by the Federal Highway Administration [SHRP2 C10B 2014, Khani et al. 2013, and Khani et al. 2014a]. The model has been tested in applications in Sacramento, CA, San Francisco, CA, Portland, OR, and Austin, TX and its integration with other DTA and activity-based models (ABM) has been established successfully.

The FAST-TrIPs model is divided into two submodules: assignment and simulation. Passengers are assigned given origin and destination zones and a preferred departure time at the origin or a preferred arrival time at the destination. Assignment can be done using a deterministic trip-based shortest path (TBSP) using travel time or a stochastic trip-based hyperpath (TBHP) using generalized travel costs (Khani et al. 2014b). In either case, vehicle capacity constraints can be included and a user equilibrium (all passengers taking their optimal available path) can be reached with iterations of assignment and simulation. In the passenger simulation module, boarding and alighting of passengers is simulated, along with other aspects of the trip (e.g. access, egress, waiting). Passengers may fail to board a transit vehicle, and the model can reassign passengers to alternate paths. Dwell time is also calculated as a function of passenger boardings and alightings at each stop. Travel statistics are accumulated and experienced skims can be generated

2. Methodology

To be updated.

See Khani (2013) for detailed methodology.

3. Input and Output Data

3.1 GTFS Input Files

The majority of the FAST-TrIPs input files are generated from General Transit Feed Specification (GTFS) files [Google Developers 2012]. The advantage of GTFS is that the model is easily applicable in any region and any scenario. The published files can represent the base conditions and manual adjustments can be made to existing GTFS files to represent various scenarios.

The GTFS input files that are needed include:

- calendar.txt
- trips.txt
- stop_times.txt
- stops.txt
- shapes.txt

GTFS files are provided by many agencies around the world and are often frequently updated to reflect service changes. Files can be downloaded for many transit agencies here: http://www.gtfs-data-exchange.com/agencies

3.2 Preparing FAST-TrIPs Input Files (on Network Modeling Center computers at UT Austin)

After the GTFS files are downloaded and unzipped, they should be moved to a directory on the server. From the directory where the GTFS files are located, use the following command in Linux terminal:

```
$ python /home/shared_files/FT_input_generation/ft_inputFileGenerator.py
```

When prompted, enter a date that represents transit service that is to be modeled (e.g. a weekday date, not a holiday, etc.). It is important to check the GTFS calendar file and pick a date within the range of service dates.

When prompted, enter “y” to prepare FAST-TrIPs input files. When complete, the following files are created in the current directory:

- ft_input_trips.dat
- ft_input_routes.dat
- ft_input_stops.dat
- ft_input_stopTimes.dat
- ft_input_shapes.dat

Access link and transfer files need to be generated for FAST-TrIPs. To generate access links, a zones files (ft_input_zones.dat) file is needed. The 2010 base year zones for the CAMPO model can be found here:

```
/home/shared_files/FT_SampleFiles
```

Add ft_input_zones.dat to the current directory (where GTFS files and FAST-TrIPS input files are now located). From the current directory use the following commands to generate access and transfer links:

```
$ python /home/shared_files/FT_input_generation/ft_accessLinkGenerator.py
$ python /home/shared_files/FT_input_generation/ft_transferLinkGenerator.py
```

The following files are created in the current directory:

- ft_input_accessLinks.dat
- ft_input_transfers.dat

FAST-TrIPs require demand in the form of disaggregate person trips. The demand file, ft_input_demand.dat, should be added to the current directory. Demand for the 2010 base year from the CAMPO model for various periods can be found here:

/home/shared_files/FT_SampleFiles

The final two input files to FAST-TrIPs specify different parameters for running the model (ft_input_routeChoice, ft_input_parameters). These final two files should be added to the current directory. Sample files can be found here:

/home/shared_files/FT_SampleFiles

A total of 11 FAST-TrIPs input files should be in the current directory and the model can now be run. Note, the original GTFS files will not be needed at this point; they can be either removed or saved for future reference.

3.3 Description of FAST-TrIPs Input and Output Files

Input and output files are in a tab-delimited text format. Input files have the prefix “ft_input_” and output files have the prefix “ft_output_”. Note, with the exception of ft_input_parameters.dat, all input files contain a first line header with column field names.

3.3.1 Input Files

The input files to FAST-TrIPs specify the transit network and schedule, passenger demand and parameters for the model.

Name	**ft_input_trips.dat**
Description	Contains a record for every transit vehicle trip
Columns	

tripID	ID that uniquely identifies a vehicle trip
routeId	ID that uniquely identifies a route
type	Service type: 0 - Tram, streetcar, light rail 1 - Subway, metro 2 - Rail 3 - Bus 4 - Ferry 5 - Cable car 6 - Gondola, suspended cable car
startTime	Start time of the trip
capacity	The vehicle capacity for the trip (default is 60)
shapeId	ID that defines a shape for the trip
directionId	ID that contains a binary value that indicates the direction of the trip

Name	**ft_input_routes.dat**
Description	Contains a record for every transit route
Columns	
routeId	ID that uniquely identifies a route
routeShortName	Short name of the route
routeLongName	Full name of the route
routeType	Service type: 0 - Tram, streetcar, light rail 1 - Subway, metro 2 - Rail 3 - Bus 4 - Ferry 5 - Cable car 6 - Gondola, suspended cable car

Name	**ft_input_stops.dat**

Description	Contains a record for every transit stop or station
Columns	
stopId	ID that uniquely identifies a stop or station
stopName	Name of a stop or station
stopDescription	Description of a stop or station
Latitude	Latitude of a stop or station (WGS 84)
Longitude	Longitude of a stop or station (WGS 84)
capacity	Capacity of a stop or station (use a large number if unknown)

Name	**ft_input_stopTimes.dat**
Description	Contains a record for every scheduled stop within a trip and route
Columns	
tripId	ID that uniquely identifies a trip
arrivalTime	Arrival time at a specific stop for a specific trip on a route in HHMMSS format measured from midnight. For trips that span multiple dates, the time should be entered as a value greater than 240000 (e.g. 250500 for 01:05:00 AM)
departureTime	Depature time from a specific stop for a specific trip on a route in HHMMSS format measured from midnight. Follow same convention as arrivalTime.
stopId	ID that uniquely identifies a stop
sequence	Sequence number of a specific stop within a trip. The first stop sequence is 1 and subsequent stops in the trip a sequentially numbered.

Name	**ft_input_shapes.dat**
Description	Contains a record for a single shape point in a single shape that collectively describes the path transit vehicles take on their trips

Columns	
shapeId	ID that uniquely identifies a shape
latitude	Latitude of a shape point (WGS 84)
longitude	Longitude of a shape point (WGS 84)
sequence	Associates the latitude and longitude of a shape point with its sequence order along a shape
distTraveled	Distance from the first shape point as real distance along the route in feet (optional)

Name	**ft_input_zones.dat**
Description	Contains a record for zone (TAZ)
Columns	
ID	ID that identifies the zone
Lat	Latitude of the zone centroid (WGS 84)
Lon	Longitude of the zone centroid (WGS 84)

Name	**ft_input_accessLinks.dat**
Description	Contains a record for each stop that can be accessed from each zone
Columns	
TAZ	ID that identifies the zone
stop	ID that identifies the stop
dist	Walking distance in miles from the TAZ centroid to the stop
time	Walking time in minutes from the TAZ centroid to the stop

Name	**ft_input_transfers.dat**
Description	Contains a record for location a transfer can be made between two stops

Columns	
fromStop	ID of stop for arriving vehicle
toStop	ID of stop for connecting vehicle
dist	Walking distance in miles between the two stops
time	Walking time in minutes between the two stops

Name	**ft_input_demand.dat**
Description	Contains a record for every passenger to be routed during assignment
Columns	
passengerId	ID that uniquely identifies a passenger
OrigTAZ	Origin zone (TAZ)
DestTAZ	Destination zone (TAZ)
mode	Transit mode (3 - walk to transit)
timePeriod	Demand time period (e.g. AM, PM, OP)
direction	Binary value indicating the passenger's direction (e.g. direction = 1 for outbound of home in AM, direction = 2 for inbound to home in PM).
PDT/PAT	Preferred departure time (if direction = 2)/preferred arrival time (if direction = 1) in minutes after midnight

Name	**ft_input_routeChoice.dat**
Description	Contains values for parameters used in the route choice model in the stochastic assignment model
Rows	
1	Header
2	Weight of in-vehicle time
3	Weight of waiting time

4	Weight of access walking time
5	Weight of egress walking time
6	Weight of transfer walking time
7	Transfer penalty in minutes
8	Weight of schedule delay (0 - no penalty)
9	Fare in dollars per boarding (with no transfer credit)
10	Value of time in dollars per hour (generic parameter)
11	Dispersion parameter in the logit function (higher values result in less stochasticity; must be nonnegative; if unknown use a value between 0.5 and 1)
12	Binary flag for including (1) or excluding (0) the vehicle capacity constraint

```
Value       :Parameter (unit: min)
1           :in-vehicle time
2.78        :waiting time
7.23        :access walking time
7.23        :egress walking time
7.23        :transfer walking time
78.07       :transfer penalty in minute
1           :schedule delay
1           :base fare in dollar per boarding (with no transfer credit)
5.85        :VOT in Dollar per hour (generic parameter)
0.5         :theta
1           :Capacity Constraint (0 or 1)
```

Figure 1 Sample ft_input_routeChoice.dat file.

Name	**ft_input_parameters.dat**
Description	Contains parameters and settings for the model, including the assignment type and outputs to generate
Rows	
1	Maximum number of iterations to remove capacity violations. When the transit system is not crowded or when capacity constraint is relaxed the model will terminate after the first iteration

2	Assignment type. 0 - No Assignment (only simulation, given paths in the input) 1 - Deterministic Assignment 2 - Stochastic Assignment
3	Simulation flag. It should be on for iterative assignment. In a one shot assignment with simulation flag off, the passengers are assigned to paths but are not loaded to the network.
4	Passenger trajectory output flag. Passengers' path and time will be reported if this flag is on. Note that the simulation flag should be on for passengers' time.
5	Path time-window. This is the time in which the paths are generated. E.g., with a typical 30 min window, any path within 30 min of the departure time will be checked.
6	Integration with DTA flag. As of now, it should be off and the schedule is replaced by DTA results in the ft_input_stopTimes.dat.
7	Skim flag. This is specific to the travel demand models (not working in this version)
8	Beginning of the time period for which the skim is required.
9	End of the time period for which the skim is required.

3.3.2 Output Files

Name	**ft_output_loadProfile**
Description	Contains a record for every stop in every trip from assignment
Columns	
routeId	ID that uniquely identifies a route
shapeId	ID that uniquely identifies a shape
tripId	ID that uniquely identifies a vehicle trip
direction	ID that contains a binary value that indicates the direction of the trip
stopId	ID that uniquely identifies a stop or station

traveledDist	Distance from the first stop in the trip as real distance along the route in feet
departureTime	Departure time of the vehicle from the stop (scheduled or simulated, if integrated with DTA model) in minutes after midnight
headway	Time between subsequent vehicles in the same route and direction
dwellTime	Dwell time of vehicle at stop in seconds
boardings	Number of boarding passengers to the vehicle at the stop
alightings	Number of alighting passengers from the vehicle at the stop
load	Passenger load onboard vehicle at time of departure

Name	**ft_output_passengerPaths.dat**
Description	Contains a record for every passenger describing the assigned path
Columns	
passengerId	ID that uniquely identifies a passenger
mode	Transit mode (3 - walk to transit)
originTaz	Origin zone (TAZ)
destinationTaz	Destination zone (TAZ)
startTime	Time passenger departs from origin (same as in ft_input_demand.dat if PDT) in minutes after midnight
boardingStops	List of stops passenger uses to board (prefix “s” added to stop ID)
boardingTrips	List of trips passenger boards (prefix “t” added to trip ID)
alightingStops	List of stops passenger uses to alight (prefix “s” added to stop ID)
walkingTimes	List of walking times (access time, any transfer times, and egress time)

passengerId	mode	originTaz	destinationTaz	startTime	boardingStops	boardingTrips	alightingStops	walkingTimes
1003614	3	188	361	627.65	s3829,s5859	t1341866,t1349495	s5225,s5863	9.51,1.00,4.68
1003620	3	188	374	390.04	s3829,s2727	t1341853,t1349834	s2727,s2738	9.51,0,2.80
1003621	3	188	376	390.04	s3829,s2727	t1341853,t1349834	s2727,s501	9.51,0,2.72
1003622	3	188	377	390.04	s3829,s2727	t1341853,t1349834	s2727,s2738	9.51,0,4.68
1003623	3	188	378	528.47	s3829,s2727	t1341857,t1349831	s2727,s5950	9.51,0,2.19
1003624	3	188	379	429.47	s3829,s2727	t1341854,t1349826	s2727,s2738	9.51,0,7.95
1003625	3	188	381	495.47	s3829,s2727	t1341856,t1349829	s2727,s2612	9.51,0,6.85
1003627	3	188	405	561.47	s3829,s2727	t1341858,t1349832	s2727,s5219	9.51,0,3.47
1003634	3	188	1413	390.04	s3829,s2727	t1341853,t1349834	s2727,s501	9.51,0,6.59
1003755	3	190	361	484.62	s2727	t1349828	s495	5.38,4.98
1003756	3	190	361	454.62	s2727	t1349835	s495	5.38,4.98
1003757	3	190	362	534.62	s2727	t1349830	s4094	5.38,6.47
1003758	3	190	362	604.62	s2727	t1349815	s4094	5.38,6.47

Figure 2 Sample ft_output_passengerPaths.dat file.

Name	**ft_output_passengerTimes.dat**
Description	Contains a record for every passenger describing the simulated timing of the assigned path
Columns	
passengerId	ID that uniquely identifies a passenger
mode	Transit mode (3 - walk to transit)
originTaz	Origin zone (TAZ)
destinationTaz	Destination zone (TAZ)
startTime	Time passenger departs from origin (same as in ft_input_demand.dat if PDT) in minutes after midnight
endTime	Time passenger arrives at destination in minutes after midnight
arrivalTimes	List of times for when passenger arrives at a stop (initial boarding stop and transfer stops, if any)
boardingTimes	List of times when passenger boards at boarding stops
alightingTimes	List of times when passenger alights at alighting stops
travelCost	Travel cost (generalized cost) of passenger's path (units: minutes)
reliabilityBudget	Not used

passengerId	mode	originTAz	destinationTaz	startTime	endTime	arrivalTimes	boardingTimes	alightingTimes	travelCost
1003614	3	188	361	627.65	703.66	637.15,673.00	637.16,683.00	672.00,699.00	287.08
1003620	3	188	374	390.04	445.63	399.53,403.00	399.55,410.00	403.00,442.83	243.37
1003621	3	188	376	390.04	444.26	399.53,403.00	399.55,410.00	403.00,441.55	241.51
1003622	3	188	377	390.04	447.5	399.53,403.00	399.55,410.00	403.00,442.83	256.96
1003623	3	188	378	528.47	603.18	537.96,542.00	537.98,560.00	542.00,601.00	278.27
1003624	3	188	379	429.47	497.61	438.96,443.00	438.98,450.00	443.00,489.66	288
1003625	3	188	381	495.47	558.91	504.96,509.00	504.98,510.00	509.00,552.06	265.77
1003627	3	188	405	561.47	623.46	570.96,575.00	570.98,580.00	575.00,620.00	250.39
1003634	3	188	1413	390.04	448.13	399.53,403.00	399.55,410.00	403.00,441.55	269.49
1003755	3	190	361	484.62	528.83	489.98	490	523.86	119.07
1003756	3	190	361	454.62	498.83	459.98	460	493.86	119.07
1003757	3	190	362	534.62	581.23	539.98	540	574.76	130.74
1003758	3	190	362	604.62	647.23	609.98	610	640.76	126.74

Figure 3 Sample ft_output_passengerTimes.dat

Name	**ft_output_runStatistics.dat**
Description	Contains a summary of the assignment (and simulation) including the total number of passengers assigned, total number of passengers who complete their trip in the simulation, and the run times of FAST-TrIPs.

4. Running FAST-TrIPs

4.1 Running FAST-TrIPs on Windows

Compile the code using a C++ compiler and create an executable file. Copy the executable file to the folder where all input files are located, and run it from there.

4.2 Running FAST-TrIPs on Linux (on NMC computers at UT Austin)

To compile FAST-TrIPs on Linux, use the following command in Linux terminal:

```
$ g++ /home/shared_files/FAST_TrIPs/FAST_TrIPs.cpp -O2 -o "/destinationfolder/outputfile.out"
```

The destination folder is where your input files are located. Alternatively, you can navigate to your destination folder and specify the output file name only. For example, if your input files are in "/home/shared/test", you can either go to the test folder and use:

```
$ g++ /home/shared_files/FAST_TrIPs/FAST_TrIPs.cpp -O2 -o FT.out
```

or from any location use:

```
$ g++ /home/shared_files/FAST_TrIPs/FAST_TrIPs.cpp -O2 -o /home/shared/test/FT.out
```

To run FAST-TrIPs, simply run the output file in the directory where the input files are. In the above example, you should use:

```
$ ./FT.out
```

After the model run completes, the output files will be written in the same directory where the inputs are. *Do not* run the model in the "/home/shared_files" directory.

5. References

Google Developers (2012). *What is GTFS? - Transit*. Available at https://developers.google.com/transit/gtfs. Accessed August 2014.

Khani, A., Sall, E., Zorn, L., & Hickman, M. (2013). Integration of the FAST-TrIPs Person-Based Dynamic Transit Assignment Model, the SF-CHAMP Regional, Activity-Based Travel Demand Model, and San Francisco's Citywide Dynamic Traffic Assignment Model. In *Proceedings of the 92nd Annual Meeting of Transportation Research Board*, Washington DC.

Khani, A., Bustillos, B., Noh, H., Chiu, Y. C., & Hickman, M. (2014a). Modeling Transit and Intermodal Tours in a Dynamic Multimodal Network. In *Transportation Research Board 93rd Annual Meeting* (No. 14-5567).

Khani, A., Hickman, M., & Noh, H. (2014b). Trip-Based Path Algorithms Using the Transit Network Hierarchy. Networks and Spatial Economics, 1-19.

Khani, A. (2013). Models and Solution Algorithms for Transit and Intermodal Passenger Assignment (Development of FAST-TrIPs Model).

SHRP 2 C10(B) (2014). Partnership to Develop an Integrated Advanced Travel Demand Model with Mode Choice Capability and Fine-Grained, Time-Sensitive Networks. Transportation Research Board. Available at http://apps.trb.org/cmsfeed/TRBNetProjectDisplay.asp?ProjectID=2828. Acesseed August 2014.