

Freight and Fuel Transportation Optimization Tool Technical Documentation

FTOT 2020.4.1 Public Release Version

March 30, 2021 DOT-VNTSC-FAA-19-06 Rev. 6.1

DOT/FAA/AEE/2020-01 Rev. 6.1

Prepared for:

Federal Aviation Administration
Office of Environment and Energy
Washington, DC



Notice

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for the contents or use thereof.

The United States Government does not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the objective of this report.

REPORT DOCUMENTATION PAGE			<i>Form Approved</i> <i>OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE March 30, 2021		3. REPORT TYPE AND DATES COVERED Document, 1/2021-3/2021
4. TITLE AND SUBTITLE Freight and Fuel Transportation Optimization Tool Technical Documentation: FTOT 2020.4.1 Public Release Version				5a. FUNDING NUMBERS FB48CS/TG287 and TG291 FB48CT19/TJ035 and TJ036 FB48CW20/TK648 and TK649 FB48CW20/UK648 and UK649
6. AUTHOR(S) Kristin C. Lewis, Matthew N. Pearlson, Alexander Oberg, Olivia Gillham, Scott Smith, Gary Baker, Michelle Gilmore, Amro El-Adle, Mark Mockett, Kirby Ledvina, Kevin Zhang				5b. CONTRACT NUMBER 693KA9-18-N-00022
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Volpe National Transportation Systems Center, 55 Broadway, Cambridge, MA 02142				8. PERFORMING ORGANIZATION REPORT NUMBER DOT-VNTSC-FAA-19-06 Rev 6.1
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Federal Aviation Administration, 800 Independence Ave, SW, Washington, DC 20591				10. SPONSORING/MONITORING AGENCY REPORT NUMBER DOT/FAA/AEE/2020-01 Rev 6.1
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT				12b. DISTRIBUTION CODE
13. ABSTRACT (Maximum 200 words) Volpe created the Freight and Fuel Transportation Optimization Tool (FTOT) to support the FAA, DOE, and the U.S. Navy's Office of Naval Research to assess optimal transport options for freight and fuel supply chains. FTOT is a flexible scenario-testing tool that optimizes the transportation of materials for future energy and freight scenarios. FTOT models and tracks commodity-specific information and can take into account conversion of raw materials to products (e.g., crude oil to jet fuel/diesel) and fulfillment of downstream demand. This report documents the components and functionality of the FTOT public release 2020.4.1 service pack. This documentation is updated quarterly and available on the Public FTOT GitHub repository: https://github.com/VolpeUSDOT/FTOT-public				
14. SUBJECT TERMS Freight, fuel, geospatial information systems, GIS, network analysis, transportation optimization.				15. NUMBER OF PAGES 48
				16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT	

Table of Contents

1	CHANGE LOG IN VERSION 2020.4.1	7
1.1	CHANGE LOG FROM VERSION 2020.4.....	7
2	INTRODUCTION.....	8
2.1	FTOT Use Cases	9
3	FTOT STRUCTURE AND COMPONENTS	11
3.1	Key software modules in FTOT	13
3.2	FTOT GIS Network.....	13
3.2.1	Data Sources	14
3.2.2	Data Preparation	15
3.2.3	Configuration of Facilities	17
3.2.4	Exporting the Data	18
3.3	NetworkX	18
3.4	Optimization	19
3.4.1	Routing and Optimization Factors	20
3.4.2	Candidate Generation Option in Optimizer	24
3.4.3	End-to-end Source Tracking Option in Optimizer	26
4	INSTALLING REQUIRED PROGRAMS.....	26
5	FTOT INPUTS	28
5.1	XML Schema	28
5.2	Configuring FTOT Scenarios in the XML File	29
5.3	Facility Data Input Files	33
5.4	Facility Geodatabase (GDB)	34
5.5	Schedule Input File.....	34
5.6	Commodity Mode Input File	35
5.7	Default Data/Input Parameters	36
6	FTOT OUTPUTS.....	39
6.1	Reports.....	39
6.2	Graphics Dashboard.....	40
6.2.1	Tableau Scenario Comparison Dashboard	41

6.3	Maps.....	41
7	SETTING UP A SCENARIO	43
8	RUNNING A SCENARIO.....	44
9	SUPPLEMENTARY TOOLS	44
10	TROUBLESHOOTING.....	45
10.1	Known Bugs and Issues	47
11	ACKNOWLEDGEMENTS	48
12	REFERENCES	48

Table of Figures and Tables

Figure 1: The four use cases for which FTOT is designed.	9
Figure 2: Generalized supply chain structure in FTOT showing origins, processors, and destinations, with multiple commodities flowing along the various steps in the process. Commodities can be facility-type specific or can share facilities (e.g., all types of crude oil can flow into any refinery, versus jet fuel only permitted to flow to airports, diesel to other types of demand centers). The number of steps in the supply chain, commodities, processor types, and availability/schedule can all be expanded.....	10
Figure 3 - Analytical tool data flow schematic showing the key components/roles of each component of the FTOT	12
Figure 4: Default multimodal network feature classes included in base FTOT network. The user will add feature classes for origin, processor, and destination facilities.	14
Figure 5 - FTOT multimodal transportation network.....	17
Figure 6: Attributes of the facilities and network segments in FTOT that the user can adjust to build a scenario.....	28
Figure 7 - Split screen practice for running FTOT scenario. The command line is on the left, and the scenario opened in the Windows Explorer is on the right.....	43
Table 1. Example of schedule input file containing two 7-day schedules.	35
Table 2 - commodity_mode.csv for two commodities: crude oil, and a generic petroleum product (pet_prods). Crude oil is only allowed to flow on the crude pipeline network (pipeline_crude), and not allowed in the petroleum products pipeline (pipeline_prod). Similarly, the generic petroleum product (pet_prods, e.g. gasoline, diesel, jet fuel) commodity is not allowed to flow on the pipeline_crude network, but is allowed to flow on the pipeline_prod network.....	36
Table 3: Modal cost units and default values in FTOT.	36
Table 4: Modal impedance factors applied to the base modal cost based on modal link category (road) or existing flow bin (rail, waterway).....	36

1 CHANGE LOG IN VERSION 2020.4.1

The 2020.4.1 service pack release adds the ability to turn the network presolve step included in the 2020.4 release on or off through an optional network density reduction (NDR) function. The functionality is controlled by the `NDR_On` parameter in the scenario configuration file; the default is for the step to be disabled. Including the parameter in the scenario configuration file and setting it to `True` enables the network presolve step described below. While the presolve enhances performance, it does not guarantee optimality of the solution found. The NDR function reduces run time for computationally intensive scenarios, but in its current form is best used for either unimodal scenarios (e.g., just road) or in cases where the goal of the analysis is high-level results and relative information across scenarios rather than detailed routing information. Additionally, the service pack release includes improvements to the shortest path algorithm used by the network presolve step. Finally, small fixes to handle unit conversion were made.

1.1 CHANGE LOG FROM VERSION 2020.4

The 2020.4 release includes a network presolve step that identifies links used in the shortest path between sources and targets (e.g. `RMP->Dest`, `RMP->Proc`, `Proc->Dest`). It is enabled by default for all scenarios except candidate generation and capacity. The presolve reduces the complexity of the optimization and enhances performance. Multi-commodity input and Processor facilities were enhanced to constrain the maximum capacity of the facility. Additionally, the release also includes changes throughout for forward compatibility with Python 3.x in the future. Note however, that 2020.4 still requires Python 2.7 because of the ArcPy module dependency. Finally, several miscellaneous fixes to improve the quality of the code base.

GENERAL

- Removed remaining references to `xtot_objects` module.
- Added Python3 forward compatibility and code cleanup
- Added compatibility for ArcGIS basic license for certain operations in `ftot.py`, `ftot_postprocess.py`, and `ftot_routing.py`
- The network is presolved for each appropriate source and target (e.g. Origin-Destination pair) using the shortest path algorithm in the NetworkX module. The links used in the shortest path are stored in the `shortest_path` table in the `main.db` and used to speed up edge creation in the O1 step and solving the optimization problem in the O2 step.

PULP MODULE

- The optimization problem only includes edges created for network links identified in the shortest path network presolve, except for scenarios that include candidate generation or capacity.
- Multi-commodity inputs for processors are properly constrained using a `max_processor_input` field in the `proc.csv` input files.

2 INTRODUCTION

Volpe created the Freight and Fuel Transportation Optimization Tool (FTOT) to support the FAA, DOE, and the U.S. Navy's Office of Naval Research to assess optimal transport options for freight and fuel supply chains. FTOT is a flexible scenario-testing tool that optimizes the transportation of materials for future energy and freight scenarios. FTOT models and tracks commodity-specific information and can take into account conversion of raw materials to products (e.g., crude oil to jet fuel and diesel) and the fulfillment of downstream demand.

FTOT's network analysis is based on a unique, multimodal, flowable Geographic Information System (GIS) network that assigns costs, capacity and existing flow information, and weighting factors to each link. It takes origin and destination (OD) data, identifies best routes among the origins and destinations, and then optimizes flow patterns by minimizing overall scenario-wide costs. The costs include transport costs as well as user-defined weights and penalties (e.g., to prefer larger roadways over smaller), transloading (mode switching) costs, and, if necessary, capital costs to build new processing locations in between the origins and destinations to convert/refine raw material. FTOT encourages flows toward common freight paths while also providing the option of taking into account modal flow capacity and capacity availability (capacity minus existing flows) by mode, based on user needs. Outputs include commodity, facility, and link-level information on flows and demand fulfillment, maps, and summary graphics.

Based on variations in infrastructure (e.g., of the transportation network), cost elements, conversion factors, and origins, destinations and waypoints in the scenario, among other parameters, the user is able explore how variants of future scenarios will impact optimal multimodal freight transport costs, transport vehicle/vessel/pipeline demand, and emissions.

In summary, FTOT provides a "best case" transportation outcome of the scenario that can be used as a screening tool for feasible scenarios and as a benchmark for comparing against real world outcomes.¹

The first public release of FTOT (Beta Version 2019.1) was made available on the GitHub code hosting site at (<https://github.com/VolpeUSDOT/FTOT-Public>) on June 24, 2019. FTOT Beta v. 2019.1 is considered an "experts' release," requiring the user to have some knowledge of Python and ArcGIS to run. It enables the user to analyze scenarios using either the default network and parameters or user-defined details. The Beta software and documentation was sent to 12 initial testers outside of the Volpe organization. Feedback was obtained from the testers and used to improve the tool for a wider general public release. The first official full version of FTOT (2019.3) was released in October 2019. Since then, FTOT versions have been released quarterly. FTOT version 2020.4.1 incorporates numerous bug fixes, enhancements, and updates as noted in the change log above.

¹ Note: Optimal scenario results can differ from real-world patterns due to complexities in transport availability, contracting mechanisms, habits, and nuances in demand and pricing, among other system dynamics.

2.1 FTOT Use Cases

One of the unique aspects of FTOT is that it focuses on transport for supply chains as opposed to a single commodity. Generally these take the form of:

Origins (+ outgoing storage) → Processor facilities type 1 (pass through or conversion) (+ incoming/outgoing storage) → Processor facilities type...x → Destinations (+ incoming storage).

The four key use-cases are described in Figure 1.

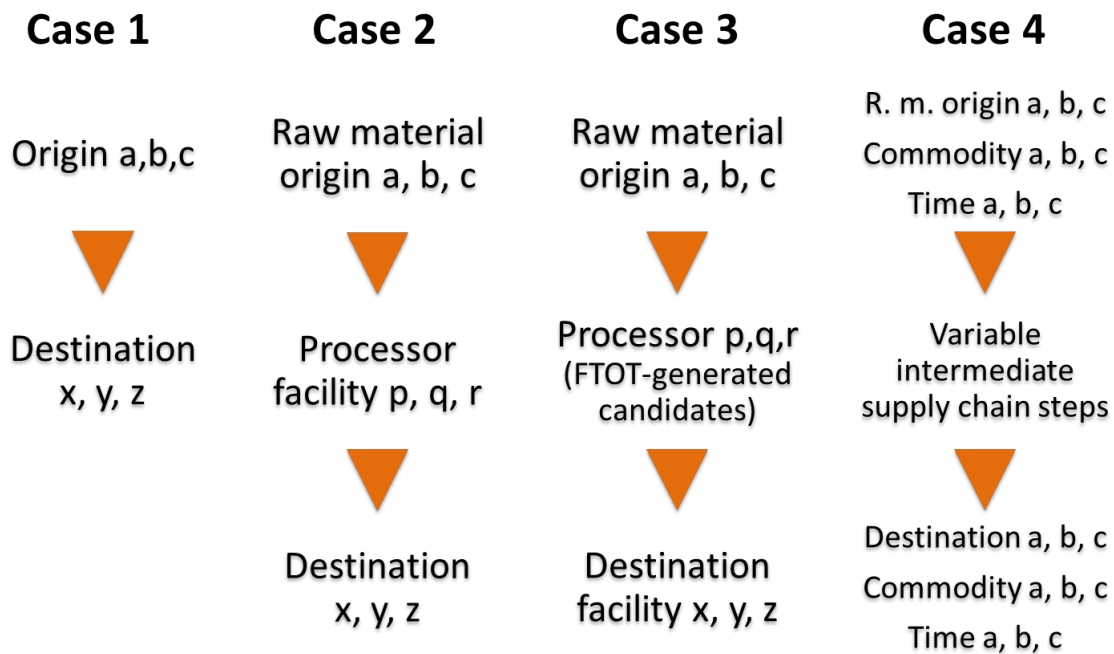


Figure 1: The four use cases for which FTOT is designed.

Multiple commodities can be flowed, converted, and tracked through the analysis to enable optimization of complex supply chains. These commodities can share origins, processors and destinations or can be canalized to separate processors or destination types depending on commodity and designated facility compatibilities. Furthermore, FTOT includes the option to schedule facility availability down to the daily level. The supply chain can be expanded by any number of steps, commodities, processor type, and time (e.g., facility availability, route availability). The generalized supply chain structure used in FTOT is shown in Figure 2.

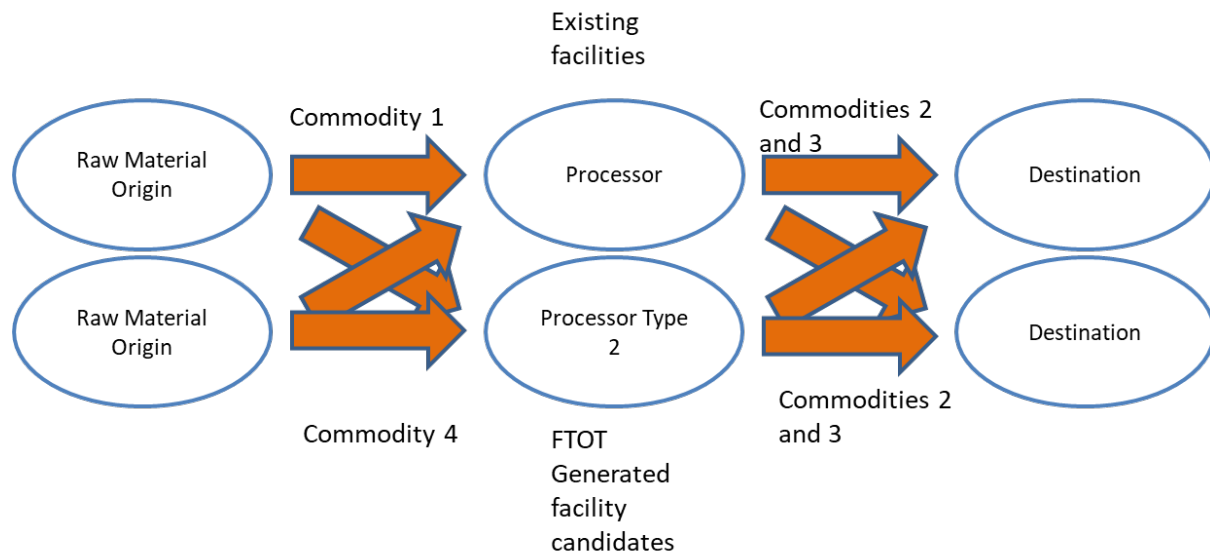


Figure 2: Generalized supply chain structure in FTOT showing origins, processors, and destinations, with multiple commodities flowing along the various steps in the process. Commodities can be facility-type specific or can share facilities (e.g., all types of crude oil can flow into any refinery, versus jet fuel only permitted to flow to airports, diesel to other types of demand centers). The number of steps in the supply chain, commodities, processor types, and availability/schedule can all be expanded.

3 FTOT STRUCTURE AND COMPONENTS

The FTOT model takes the GIS-based network, transforms it into a consumable multi-graph using an open-source network analysis tool, and feeds the multi-graph and scenario parameters into an optimization module. The optimizer selects among origins, processors and destinations and allocates flows among them to maximize commodity delivery and minimize overall cost, taking into account multiple cost elements, weightings, and constraints and described in Section 3. Figure 3 demonstrates the FTOT architecture and data flow. Installation instructions for these required software modeling tools are described in Section 4.

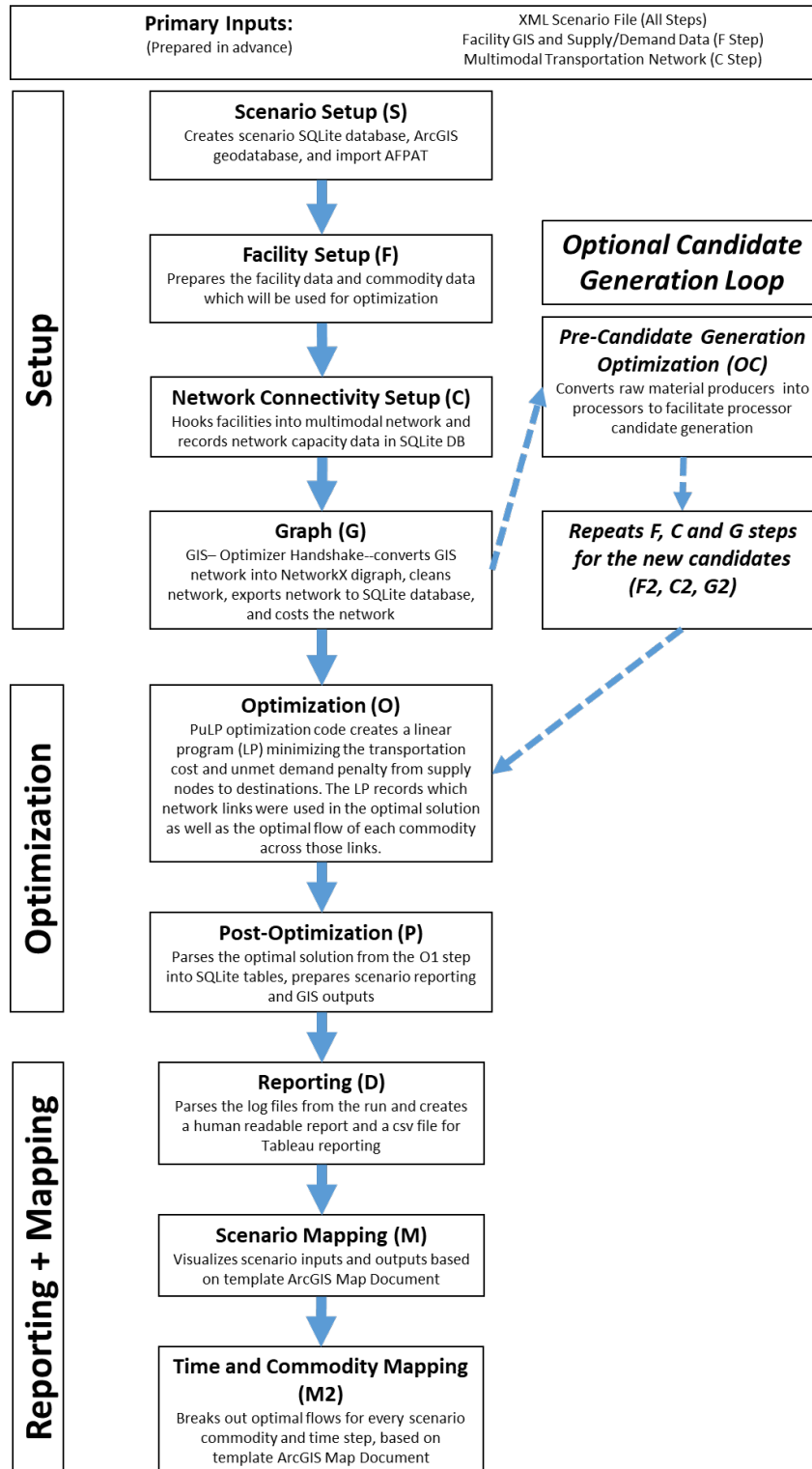


Figure 3 - Analytical tool data flow schematic showing the key components/roles of each component of the FTOT

3.1 Key software modules in FTOT

The following software tools are integral to the use of FTOT:

Python v2.7 is the scripting language supporting FTOT. FTOT relies specifically on the Esri ArcGIS installation of Python 2.7—which among many default modules, includes the Arcpy module for GIS functionality. Additional supporting Python modules are also installed on top of a base Python installation.

ESRI ArcGIS supports the GIS component of the tool. The Arcpy module is a Python-based tool that allows access to ArcGIS functionality. FTOT uses Arcpy to process the multi-modal network and facility locations defined for the scenario. Components of ArcGIS apply transport costs, impedances, capacity, and existing flows to the network. This network is then passed through NetworkX to the optimizer for analysis. After optimization, the outputs of the scenario are passed back to the GIS module to generate scenario maps.

NetworkX is used to interface between ArcGIS and the optimizer. NetworkX is a Python-based open source software package that enables network analyses. NetworkX is able to read in the scenario GIS data, and translate it into a format that is readable by the optimizer. NetworkX can be used for network flow optimization, but is not able to take into account some of the required elements in FTOT, specifically the ability to optimize using multiple paths between origin-destination pairs to allow for capacity-constrained analyses in which the optimizer will use a shortest path until capacity is reached, then route flows to the next shortest path, and so on. Starting with the 2020.4 version of FTOT, a network presolve functionality is included for analyses that do not use capacity nor candidate generation. When this functionality is enabled, the NetworkX `shortest_path` algorithm is used to identify links for use in the optimization problem. Starting with the 2020.4.1 service pack release, the functionality is made optional as specified in the scenario configuration file.

The **PuLP** optimizer is a Python-based tool that uses standard operations research algorithms to solve optimization problems. Once a problem is described in mathematical terms, PuLP can be used to find the optimal values. In FTOT, the optimization objective is to satisfy maximum demand based on least cost. More detail on the optimizer and the COIN-OR solver can be found in the sections below.

A **Tableau dashboard** is used to generate a suite of graphical outputs based on the optimal scenario solution. To view the Tableau dashboard, one can use either the free Tableau Reader or a full version of Tableau Desktop v2019.4 or higher.

3.2 FTOT GIS Network

The elements of the FTOT multimodal network include feature classes for road, rail, water, and pipeline networks. Intermodal facilities where transfers between modes occur, waterway locks, and available tariff and capacity information for crude and product pipelines are also included in the network feature classes. Each pipeline category has three feature classes associated with them: 1) pipeline feature classes represent the segments associated with the pipeline network; 2) `pipeline_trf_sgmts` feature

classes store the ordered segments associated with each tariff for which FTOT has reliable cost and origin/destination data; and 3) pipeline_trf_rts contains tariff-level information such as the cost of the movement and information on the pipeline and origin/destination pipeline stations. FTOT uses these feature classes together to limit pipeline movements to the predetermined routes that tariffs determine. Default feature classes are shown in Figure 4. Full metadata for the multimodal network can be accessed in ArcMap or ArcCatalog, part of the ArcGIS suite of software that is required to run FTOT. A summary of the data sources behind the multimodal network is provided below.

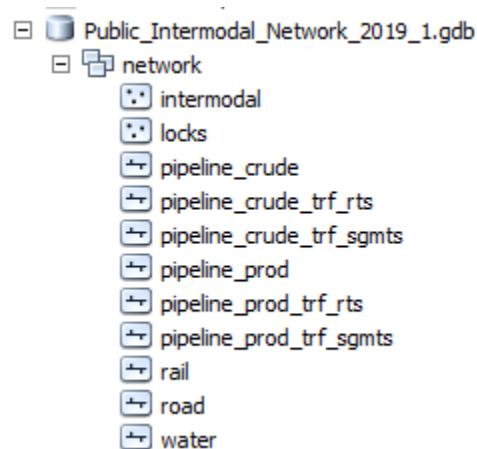


Figure 4: Default multimodal network feature classes included in base FTOT network. The user will add feature classes for origin, processor, and destination facilities.

3.2.1 Data Sources

The FTOT install includes a base multimodal transportation network composed of features and selected attributes from the following source datasets:

- Network source data / components
 - Road: FHWA Freight Analysis Framework (FAF) v. 4--2016) (accessed via the [Federal Highway Administration Office of Operations](#))
 - Rail: FRA North American Rail Network (NARN)—2018 (sent directly to Volpe by FRA). Only publicly available attributes of the NARN are included in the public release of the FTOT network.
 - Waterway: U.S. Army Corps of Engineers (USACE) Navigable Waterway Network--2018 (accessed via the [USACE Navigation Data Center](#)), modified based on additional research to exclude some non-navigable waterways.
 - Locks: USACE Locks Shapefile—2018 (accessed via the [USACE Navigation Data Center](#))
 - Crude and Petroleum Product Pipeline Network Data: Energy Information Administration (EIA) crude oil and petroleum product pipeline data- March, 2017 (accessed via [EIA Maps](#)), modified by Volpe based on station locations in a tariff dataset by LawIQ Inc. (February, 2018) and other publicly available pipeline network data.
 - Crude and Petroleum Product Pipeline Tariff Rate Data: Pipeline transportation rate and origin/destination information supplied by LawIQ Inc. For analysis purposes, all

rates have been rounded (nearest \$1 for rates from \$0-\$20; nearest \$5 for rates from \$20-\$50, and nearest \$10 for rates over \$50). If further detail is needed, the user can contact info@lawiq.com for precise tariff-level rates for over 1,000 interstate and intrastate pipelines across 100 unique commodity types and 40 jurisdictions.

- Intermodal facilities: Volpe-developed list of intermodal facilities that handle energy products based on public data from:
 - EIA [Crude Oil by Rail Dataset](#) (November, 2014)
 - USACE [Master Docks Database](#) (February, 2016)
 - Corroboration from Surface Transportation Board Waybill Sample dataset (2013)
 - Pipeline station data from LawIQ Inc. (February, 2018)
 - Volpe research/review of facility locations using satellite imagery and other research of public data sources

3.2.2 Data Preparation

While the source data for road, rail, waterway and locks are provided in formats that FTOT can leverage with little modification, FTOT's pipeline network and intermodal facilities datasets were produced especially for FTOT and require standalone processes that integrate various datasets, described below.

To prepare the FTOT pipeline network, a series of programmatic methods are used to integrate tariff cost data and origin/destination station location data from LawIQ Inc. with their corresponding pipeline segments in the Volpe-modified EIA GIS pipeline network, providing a flowable route across the pipeline network for each tariff. The tariff costs are attached to each tariff's approximate route, and these sets of routes and costs become candidates for movements of relevant commodities in FTOT, often as part of larger multimodal movements. A product is considered to flow between existing origin-destination pairs within the pipeline network (i.e., no new injection or extraction points are allowed).

The intermodal facilities dataset was created by combining two separate efforts to identify 1) agriculture-related intermodal facilities, and 2) energy-related intermodal facilities. Intermodal facilities may be added or removed from time-to-time during periodic updates of the data sources.

The agriculture-related intermodal facilities dataset was created from the Bureau of Transportation Statistics' public intermodal facility list using specific criteria and visual review of satellite imagery to correct facility locations and eliminate excess facility points. Criteria for elimination included: lack of nearby significant intermodal facility in satellite imagery, proximity to other, more accurate intermodal points, points that did not have complete data and/or points that were not applicable to the FTOT network (e.g., made at least one linkage among road, rail, waterway, and pipeline). Rail intermodal points were added to the remaining list based on movements of specific, raw-material-relevant commodity types in the Railway Waybill Sample to identify potential railway entry points for agricultural commodities, which are likely to enter the system at smaller facilities than the main intermodal facilities identified by BTS. The commodity-specific origins were incorporated into the GIS layer and duplicates

were eliminated. The final intermodal facilities layer for FTOT includes 383 unique intermodal facilities tagged specifically for the different modes that interface at each point.

The energy-related intermodal facilities were defined through a separate effort to integrate Army Corps of Engineers port facilities handling coal, crude oil and petroleum products; Energy Information Administration crude-by-rail facility data; and other facilities identified through a screening process involving Railway Waybill Sample data and manual review of satellite imagery.

FTOT includes a default national multimodal network including the road, rail, waterway, pipeline, and intermodal facility layers. The network is based on the authoritative sources and Volpe-produced datasets described above and is processed to simplify network segmentation as much as possible while retaining core required FTOT attributes—such as existing flows, capacity and other attributes helpful for determining network impedances.

For datasets not produced especially for FTOT, some additional filtering of the source network data is necessary. For example, in the FTOT network FRA's North American Rail Network is subset to the main subnetwork and major industrial leads within the continental United States. The USACE Navigable Waterway Network is subset to exclude waterways that are not navigable to a depth of nine feet (standard barge clearance). Note that water movements in FTOT are assumed to travel by barge, not ship.

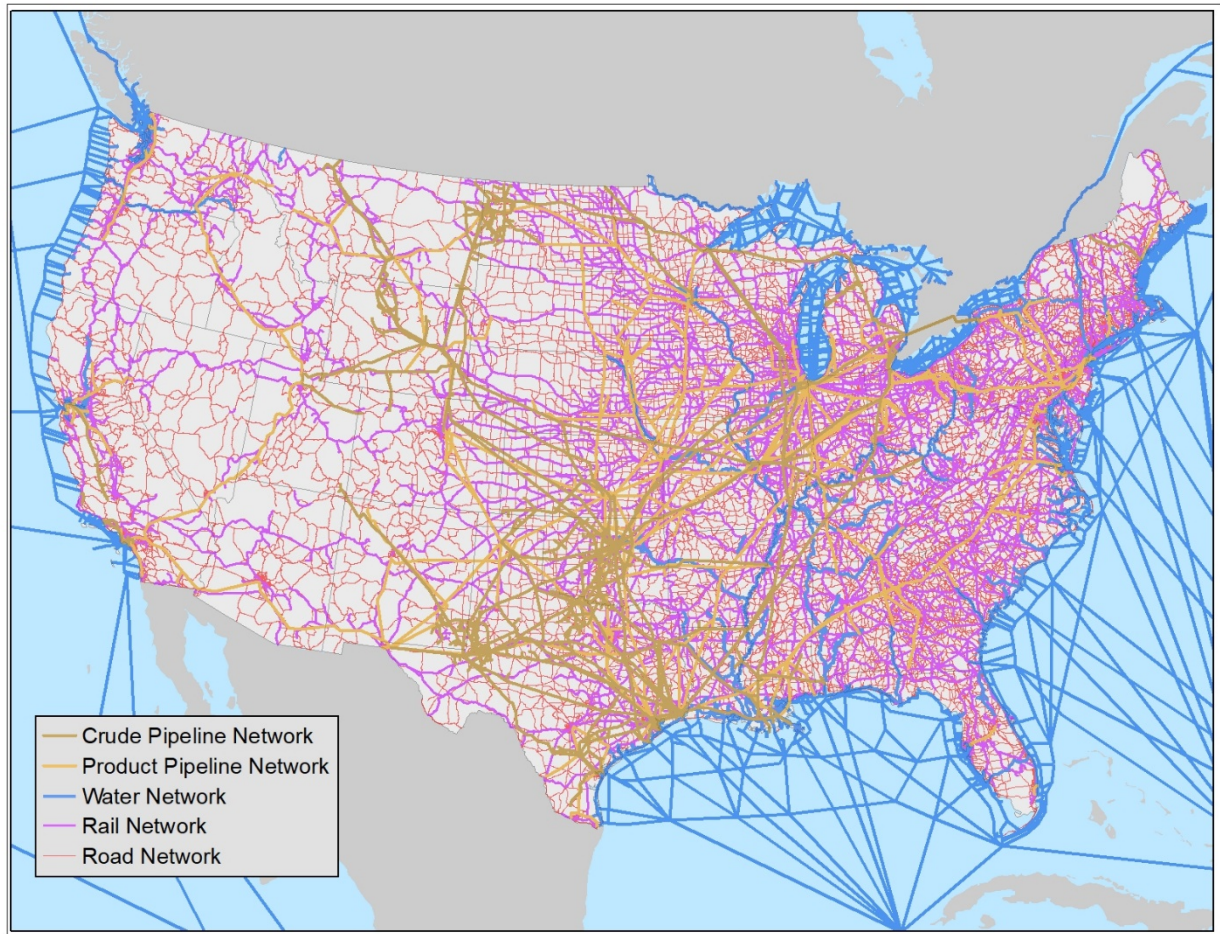


Figure 5 - FTOT multimodal transportation network

Figure 5 shows the resulting multimodal network that FTOT uses for routing analyses and display of results.

3.2.3 Configuration of Facilities

In addition to the multi-modal network, FTOT requires origination facilities to supply material in the scenario, and destinations with demand to terminate the flows. Origin facilities are by definition facilities that have an output but no input. Conversely, destinations are defined as facilities that have inputs but no outputs. Processors are an optional facility type that can be included in the supply chain. FTOT considers any facility that has inputs and outputs to be processors. Processors may process the material from one commodity to another, or simply act as a pass-through where material is aggregated or stored. During an FTOT scenario, FTOT copies user-specified facilities into the scenario network geodatabase and programmatically links them to the multimodal network. The process of connecting facilities to the network modes is known as adding artificial links. The fully connected network is stored in a feature dataset and includes the multimodal network, intermodal facilities, and origin/processor/destination facilities feature classes.

3.2.4 Exporting the Data

After the facilities are connected to the multimodal network with artificial links, the network feature classes are exported as shapefiles in preparation for the next step, converting the network into a graph. The NetworkX module, described in greater detail below, is used to create a graph of the connected network. NetworkX uses shapefiles as the geospatial input. Therefore, the feature classes for each network feature dataset are exported as shapefiles using Arcpy and stored in a temporary directory for NetworkX.

3.3 NetworkX

NetworkX is used to create a network graph from the geospatial network and facility data. Shapefiles are the required geospatial input. Each element of the network feature dataset (roads, rail, destinations, etc.) are exported as shapefiles and stored in a temporary directory. Next, NetworkX reads in each shapefile and adds it into a network graph.

Once the base graph is created, it is processed with the following procedure to ensure flows in both directions are enabled for certain modes (e.g. road, water, and rail) and restricted for other modes (e.g. pipeline). First, a copy of the network is duplicated. The duplicate is then reversed to enable flow in opposite directions. The duplicate-reverse graph and the original graph are then joined back together, after which the network edges are inspected individually and cleaned. Cleaning includes:

- Deleting duplicate artificial links:
 - Origin facilities should only have links flowing out.
 - Destination facilities should only have links flowing in.
- Deleting reversed pipeline links (pipeline only flows in one direction).
- Node IDs are converted from latitude longitude coordinates to integer values, which simplifies labeling and graph operations.
- Network costs are calculated based on link attributes and user-defined inputs. Two types of network costs are recorded. First, the dollar cost is based on the mileage and mode. Second, a routing cost is stored that includes impedances based on the network mode and link attributes. Impedances are used to encourage flows on main links of the network, and penalize minor link usage.

Once the network graph is cleaned, it is stored in a local SQLite database and is ready for the optimization steps.

A new feature added in 2020.4 includes the ability to presolve the network using the NetworkX shortest path algorithm for scenarios without candidate generation or capacity considerations. Starting with the 2020.4.1 service pack, this feature is optional; the default is for the presolve step to be disabled. To enable the network presolve, an additional parameter named `NDR_On` must be included in the scenario configuration file and set to `True`. (Exercise 3 of Quick Start 6 in the Quick Start documentation provides an example of this functionality.) Shortest paths are identified for each source and target location sharing a commodity. For example, RMP to Destination, RMP to Processor, and Processor to Destination

source and target pairs are solved with the shortest path algorithm. An approximated routing cost is used as the weight for each link. The modified weight is calculated as the route cost scaling factor times the length of the link in miles. Base rates for all transport modes and all phases of matter are set to the default costs for liquids specified in the FTOT default configuration (Table 3), and therefore differences in the cost of moving each phase of matter are ignored in this calculation.

The links identified in the shortest path solution are then stored in the main.db in the 'shortest_edges' table. Unlike previous versions of FTOT, the optimizer generates the linear program using only edges contained in the 'shortest_edges' table. This significantly reduces the complexity of building the problem in the O1 step, and the solution in the O2 step. Since the routing costs used in the shortest path solution are approximate, the solution offered by FTOT when the network presolve step is enabled cannot be guaranteed optimal. The network presolve step reduces run time for computationally intensive scenarios, but in its current form is best used for either unimodal scenarios (e.g., just road) or in cases where the goal of the analysis is high-level results and relative information across scenarios rather than detailed routing information. The network presolve is never run when candidates or capacity is enabled.

The following section describes the optimization: how the problem is defined and solved using the linear programming solver.

3.4 Optimization

The goal of the PuLP optimization process is to minimize the total cost of transporting enough material from the origin facilities, through the processor (if included in the supply chain), and on to the destinations to meet demand. This analysis includes factors for actual transportation costs that vary by the mode and length of the routes, weightings for route links, capital costs for building refineries, and penalties for not meeting demand. The three components of the optimal cost are described below:

- 1) The actual dollar costs of transporting the material plus additional routing costs from impedances (weightings and penalties that force the tool to favor particular desirable characteristics of the routing).
- 2) The amortized annual capital cost of building any candidate facilities that are part of the solution. If no candidate facilities are included in the scenario, this cost is equal to zero.
- 3) Destination facilities with unmet demand are also treated as a "cost" in this system by applying a penalty per unit of undelivered commodity. Without this sort of penalty, the lowest cost solution would always be to transport nothing at all.

The FTOT network flow and facility location problem can be mathematically formulated as a linear programming problem (Bertsimas and Tsiklis 1997). Solving problems of this type optimally (or near-optimally) is one focus of operations research. Computational Infrastructure for Operations Research (COIN-OR) maintains a repository of open-source software for the operations research community. This repository is the source of the PuLP ("pulp-or" 2007, pulp-or 2014) and Coin-OR Linear Programming (CLP) (COIN-OR 2007) software used for the FTOT project. PuLP is a linear programming modeler, used to set up the linear problem and then call the CLP solver. The CLP solver does the work of solving the

linear programming problem (Bertsimas and Tsiklis 1997) using the Simplex algorithm (Carreira-Perpiñán 2014).

The PuLP optimizer identifies a maximum or minimum value (in this case, total transportation, capital, and unmet demand costs) using a mathematical description of the supply chain scenario. In its application for FTOT, the PuLP optimizer takes the user specified facility information (geospatial and commodity supply/demand/processing data) and the transportation network and optimizes the paths and flows among all components. The goal of the optimization is to minimize the total cost of meeting as much destination demand as possible by utilizing all available supply and transportation modes.

3.4.1 Routing and Optimization Factors

This section describes each of the factors that are considered in the route optimization algorithm.

- 1) **Monetary costs** for each transportation option and for construction (if applicable)
 - For road, rail and waterway, this is applied as a cost per metric ton-mile for solids or per thousand gallon-mile for liquids.
 - Pipeline costs are based on tariff data described above.
 - Amortized annual capital cost for candidate processor construction is also considered if candidate processors are generated in the scenario.
- 2) **Modal flow capacity** and existing flows
 - Modes can be configured as permitted or not permitted in the scenario configuration file.
 - Capacity constraints can be enabled by mode to reroute flows if and when available capacity on the network has been exceeded. Network capacity and background volumes are estimated from existing data sources which are further described in Section 5.7.
- 3) **Weightings and penalties**
 - Unmet demand penalty
 - Short haul penalty
- 4) **Unmet demand penalty** if a destination does not receive the desired quantity of a given commodity
 - The optimizer adds a penalty (proportional to the amount of unmet demand) to the total cost of a potential solution.
 - The penalty (per unit of unmet demand) is specified by the user in the scenario configuration file.
 - If the magnitude of the penalty is low, the optimization will prioritize minimizing transportation costs; if it is high, the optimization will prioritize meeting all demand, with less sensitivity to the dollar cost of doing so.
 - This penalty is required for the optimizer to function; if there is no penalty for not meeting demand, the lowest cost solution will always be to transport nothing at all (a “no-flow

solution”). The penalty drives the model to move material rather than allow it to remain at the starting point; therefore, it represents the maximum allowable transport cost for a given unit of flow. However, the optimization will always choose the lowest cost path when possible for a given unit of material.

- The unmet demand penalty is also a common cause of errors for new scenarios. When limited geographic scope or time periods may result in a scenario that does not produce enough material to outweigh the cost of transporting material or building a candidate processing facility, a higher unmet demand penalty may be required to avoid a no-flow solution. In general, it may be necessary to raise this penalty when any other cost (e.g. rail transport) is raised, or else the optimizer will conclude that it is more optimal to transport less material. As a general guide, the unmet demand penalty will likely work best if set to be 10-50 times the average actual transportation cost. This ensures that raw material and products will be transported even over long routes. It should be noted that at very low flows, one may see non-intuitive results such as reduced utilization of raw material that is available. This is due to the optimizer electing the less expensive option of reducing flow overall instead of accumulating high transport and capital costs in addition to a still-large unmet demand penalty for a given route. Likewise, a very high unmet demand penalty may force the flow of materials in unanticipated ways. It is advised to run scenarios with multiple unmet demand penalties to explore the sensitivity of a given analysis.

5) **Short haul penalty** for short movements on the water and rail network

- A penalty for rail and water is added to the routing cost for all artificial links that connect facilities with the rail and water network. This penalty is equal to the difference between the costs of routing 100 miles on local roads and 100 miles on rail (for rail) and the difference between the costs of routing 100 miles on local roads and 100 miles on water (for water).
- The penalty does not impact the actual dollar costs as reported in the results—it is leveraged as an additional impedance within the optimization to encourage the use of road for shorter commodity flows, reflecting a reality where rail and water are typically utilized for longer commodity flows.

This penalty is currently built into FTOT code and cannot be adjusted or turned off. Future versions of FTOT may allow for this penalty to be configured by the user.

6) **Minimum flow requirements** applied to processing facilities

- A processing facility may have a defined capacity. The user provides this as a maximum input capacity for the processor. This is defined in the max_processor_input field of the proc.csv input file.
- The minimum capacity is set to one-half the maximum capacity, and only applies if the processor is in use. That is, even when a minimum is defined the flow is allowed to be zero.

7) **Network impedances** to avoid unrealistic shortest path routes on road, rail, and water segments

- Impedances encourage flow over highly used paths and discourage flow over paths where freight traffic is less common.
- Like penalties, impedance weights are added to the routing costs and used for the optimization. They are distinct from the dollar cost that FTOT reports and summarizes.
- In the scenario configuration file, the user can adjust the impedance weights on each type of segment in the network.
- See Section 5.7 for a description of FTOT’s default impedances.

The number of impedance categories and their definitions depend on the data available for the particular mode—for example, in the road network impedances are based on roadway functional class and in the water network impedances are based on actual freight flows. For the rail network, while the internal (non-public) network relies on actual freight tonnage data from the railroads to set impedances, the public network relies on a simpler set of categories based on whether or not the rail segment is part of the Strategic Rail Corridor Network (STRACNET) and whether or not it is owned or flowed on by Class 1 freight railroads.

8) **Restrictions on distance a commodity can travel** if a commodity has a maximum allowable transport distance

- Because some commodities have a maximum allowable transport distance, the optimization problem is set up so that potential routes are built link-by-link from each source facility; total distance travelled from the source facility to a given link is thus tracked.
- When the maximum allowable transport distance is reached, no further links can be added to the route on which that particular commodity travels from that particular source facility.

These costs and weightings are translated to mathematical decision variables and coefficients as follows:

<u>Variable</u>	<u>Explanation</u>
X_{iabjab}	Flow, in units/time period, from origin facility i_{ab} to origin facility storage j_{ab} , where a is the commodity produced and b is the time period of production ²
$X_{jabja(b+1)}$	Storage of commodity a at origin facility storage location j from time period b to time period $b+1$, in units/time period
X_{jabkab}	Flow, in units/time period, from origin facility storage j_{ab} to processing facility k_{ab} where a is the commodity being transported and b is the current time period
$X_{kabmabc}$	Flow, in units/time period, from processing facility k_{ab} to destination storage m_{abc} where a is the commodity that was processed, b is the current time period, c is the fuel being transported
$X_{mabcma(b+1)c}$	Storage at destination storage location m from time period b to time period $b+1$ of commodity c , produced from commodity a , in thousand gallons/time period
$X_{mabcnabc}$	Flow, in units/time period, from destination storage m_{abc} to destination n_{bc} where a is the commodity that was processed, b is the current time period, c is the fuel being transported

² “units” are defined by the user in the scenario configuration file. FTOT is able to use a wide variety of units and convert between units. Supported units are detailed in the Pint python module documentation.

<u>Variable</u>	<u>Explanation</u>
u_{bnc}	Unmet demand of commodity c at destination n during time period b, in units/time period
E_{kabc}	Excess commodity remaining at processing facility k_{ab} of commodity c, where a is the commodity that was processed, b is the current time period. Non-negative. Counts toward upper and lower bounds on production, but is not used to fulfill demand.
y_k	0 – 1 variable: 1 if processing facility k is used, 0 otherwise

<u>Coefficient</u>	<u>Explanation</u>
U_{Piab}	Upper bound on flow of commodity a out of origin facility i_{ab} during time period b, in units/time period
C_{iabjab}	Transportation cost, in \$ / unit, to flow commodity a during time period b from origin facility storage j_{ab} to processing facility k_{ab}
F_{Bk}	Fixed cost, converted to \$ / year, to build the processing facility k
U_{Bk}	Upper bound on flow out of processing facility k, in thousand gallons / time period
L_{Bk}	If processing facility j is used, lower bound on flow out of processing facility j, in thousand gallons / year
$C_{kabmabc}$	Transportation cost, in \$ / thousand gallon, from processing facility k to destination storage m, of fuel c in time period b processed from commodity a
S_{Bac}	Conversion factor at the processing facility: inbound thousand gallons or tons of commodity a x S_{Bac} = outbound thousand gallons of commodity c
U_{nc}	Penalty (\$ / thousand gallon) for not meeting demand at destination n for fuel c for any time period
D_{bnc}	Demand, in units / time period, at destination n of fuel c during time period b
M_x	Maximum flow (capacity) in units / time period on edge ³ r

Then the problem for FTOT analysis is mathematically stated as follows:

$$\text{Minimize annual cost} = \sum_{jabkab} C_{jabkab} x_{jabkab} + \sum_k (F_{Bk} y_k) + \sum_{kabmabc} C_{kabmabc} x_{kabmabc} + \sum_{bnc} (U_{nc} u_{bnc})$$

Subject to the following:

³ A route between facilities can be composed of multiple edges, each of which may have a distinct maximum capacity

Constraint	Explanation
(1) For each processing facility k_{ab} , $S_{Bac} \sum_{jabkab} X_{jabkab} - \sum_{kabmabc} X_{kabmabc} = E_{kabc}$	Flow must be conserved at each stage relative to time and commodity (with the appropriate conversion factors); any excess is tracked.
(2) For each processing facility k_{ab} , $y_k U_{Bk} - \sum_{kabmabc} X_{kabmabc} - E_{kabc} \geq 0$	If a processing facility is used, flow cannot exceed the upper bound during any time period, across all commodities, including excess. Note that if the processing facility is not used ($y_k = 0$), this constraint requires the flow out of the processing facility be 0.
(3) For each processing facility k_{ab} , $y_k L_{Bk} - \sum_{kabmabc} X_{kabmabc} - E_{kabc} \leq 0$	If a processing facility is used, the flow out of it over a year must exceed the lower bound, summing all commodities.
(4) For each origin facility i_{ab} , $\sum_{iabjab} X_{iabjab} \leq U_{Piab}$	Flow out of each origin facility does not exceed the origin facility upper bound, for each time period and commodity.
(5) For each destination n_c , $\sum_{mabcnbc} X_{mabcnbc} + u_{bnc} = D_{bnc}$	Unmet demand plus flow into a destination is equal to that destination's demand, for each commodity.
(6) The y variables are binary (0 or 1)	A processing facility is used, or it isn't.
(7) The x , E , and u variables are non-negative	No negative flows are permitted
(8) For each time period b and edge r , $\sum x_r \leq M_r$	For each edge and time period, total flow (sum of all x on that r) does not exceed capacity for that edge

The optimizer then uses standard linear optimization techniques such as a revised simplex algorithm to solve the mathematical description of the problem to move material from origin to destination by selecting among paths and processor options for each unit of the commodities. The specific choice of algorithm is made by the COINMP_DLL solver, as implemented by PuLP. The allocation of commodities/materials among paths and facilities is based on meeting maximum demand while minimizing the total cost, without violating the constraints on minimum and maximum flow.

3.4.2 Candidate Generation Option in Optimizer

- FTOT can generate screening-level processor candidate locations to convert raw material to commodities demanded by the destinations. Candidate generation requires the user to specify the input and output commodity relationships, as well as the facility maximum and minimum size, minimum aggregation size for generating a candidate, and the amortized capital cost as a function of size.
- In order to get a solution, the user must add the max_transport_distance field in the RMP commodity input file. This field is mandatory for FTOT runs with candidate generation.
- Routes are built up starting from origin points out to maximum feedstock distance.
- Beyond the maximum raw material transport distance, FTOT converts raw materials to end products demanded by the final destinations using the most efficient conversion process specified in the candidate input file, and flows these commodities to the final destinations as if

they were the commodity demanded. These routes are used to identify potential raw material aggregation along a lowest-cost path from the origin to the destination.

- FTOT identifies candidate nodes where sufficient raw material is aggregated on the network to support a potential facility. The minimum quantity of material can be set in the candidate processor commodity CSV file. Raw material that aggregates beyond the user defined raw material transport distance is ignored in the aggregation calculations.
- Candidate nodes are then assigned a process, and added to a candidate processor feature class.
- The facilities and connectivity steps are then run again to add the candidate processor locations into the scenario network as facilities and record the product slate and processor size.
- Optimization is then rerun and proceeds as usual.

3.4.2.1 Advanced Candidate Processor Functionality

The section covers the use of candidate processors from outside FTOT. Like other facilities, FTOT expects two pieces of information about externally-generated candidate processor facilities: the geospatial location specified in a candidate process feature class, and a facility-commodity.csv file.

- The FTOT generated candidate locations are stored in a feature class named “all_candidate_processors” within the main.gdb file. The feature class includes the ‘facility_name’ and ‘candidate’ fields. The facility name is based on the process name from the Processors_Candidate_Commodity_Data and the node_id the candidate is located near based on accumulated flow in the normal FTOT workflow. However, with user defined candidates the user is free to name the facility as they see fit. The candidate field is a short integer set to 1 to indicate it is a candidate processor facility.
- The candidate processor facility-commodity csv file is stored in the debug folder of the scenario and is named “ftot_generated_processor_candidates.csv”. It contains the appropriate processing ratio and facility size based on the aggregated flow on the network. The user should take care to specify all the required records for each candidate facility.

After the candidates are generated, FTOT reads in these facilities like any other input. They are added to the main.db and main.gdb just as the known RMPs, Processors, and Destinations. The only difference is that FTOT looks for these files in specific locations, and the optimizer includes a cost penalty for building these facilities. The user, is therefore, free to modify these files as they see fit. The following examples are provided to demonstrate how users may change candidate processors in FTOT after the candidate generation process is completed (e.g OC3).

- Modify the ftot_generated_processor_candidates.csv file in the debug directory to change the processing ratios of the facilities.
- Modify the all_candidate_processors feature class in the main.gdb to change the locations of the candidate facilities of interest. For example, adding additional candidate locations that were generated outside of FTOT. Note: the facility_name field must be unique and match a facility_name in the ftot_generated_processor_candidates.csv file. Similarly, the candidate field must be set to ‘1’ for each candidate record.

3.4.3 End-to-end Source Tracking Option in Optimizer

- When detailed information is required about exactly where material flowed, an additional step can be run after the main optimization.
- This method provides the end-to-end path of every piece of demand met. It should generally only be run after “O”, the standard optimization. It will use that solution and define more specific flow paths. Running it from scratch, instead of “O”, is possible but exponentially more time and resource intensive, and yields the same information.
This is accomplished by creating flow variables for every edge and *source facility* combination; essentially, the same commodity from different sources is treated (and tracked) as distinct commodities.

4 INSTALLING REQUIRED PROGRAMS

Current FTOT install instructions are available here: <https://github.com/VolpeUSDOT/FTOT-Public/wiki>. Before running a scenario in FTOT, the following programs must be installed:

- **ESRI ArcGIS** Version 10.6.1 or higher (Geographic Information System (GIS) Program) performs the geospatial analysis elements of the tool. The Advanced License level of ArcGIS is required. FTOT may run on earlier versions of ArcGIS dating back to v 10.1, but for full support and functionality v 10.6.1 or higher is highly recommended. Be sure to include the **Esri ArcGIS 10.6.1 base Python installation** (Version 2.7.14). This is installed alongside Esri ArcGIS and provides the majority of Python modules required for FTOT scripts to run. Special attention should be paid to this installation of Python. See the trouble-shooting section for issues with pre-installed Python environments.
- **ESRI ArcGIS 64-Bit Background Processing** is a separate installation of python and the arcpy modules. It is required for scenarios that exceed 2 GB of memory (RAM) during run time. It is highly recommended for installation and can be used for all scenarios, both large and small.
- **SQLite**—is used to store scenario variables and results. In order for the user to fully explore SQLite-generated outputs, users are encouraged to install a SQLite database browser, such as DB Browser for SQLite.
- **NetworkX** Version 2.2 is an open source python-based suite of tools for manipulating and analyzing complex networks for which many publicly available algorithms and tools have been established.
- **GDAL 2.2.4** is a required dependency for NetworkX to read in spatial input files for graph conversion.
- **PuLP** Version 1.6.8 (Open Source Python Wrapper for Optimization Solvers) is a linear programming module written in Python. In FTOT, PuLP is used to link the solvers in the Computational Infrastructure for Operations Research project (COIN-OR). The **COIN-OR** project maintains an archive of open source software for operations research, including implementations of algorithmic solvers and modeling systems. This includes a number of open

source optimization models, including a branch and cut solver (CBC) for mixed integer programming.

- **lxml** Version 4.2.5 is a python package for handling xml and is used in FTOT to parse the scenario configuration file.
- **Pint Version 0.8.1** is a python package for unit conversions and dimensionality checks. It is used throughout FTOT to harmonize user input data into a common set of units for solids and liquid freight movements.
- **Imageio version 2.5.0** is a Python package required for creating animated graphics of FTOT mapping outputs.

FTOT Version 2020.4.1 is the tool itself and can be downloaded from the Public GitHub repository [<https://github.com/VolpeUSDOT/FTOT-Public>].

5 FTOT INPUTS

FTOT scenarios require geospatial information for the network itself as well as associated attributes such as costs, impedances and weightings, capacity, movement restrictions, and schedule. The FTOT network must also include the facilities (origins, processing/waypoints, destinations) associated with the supply chain being analyzed, as well as their associated attributes such as facility minimum and maximum size, available supply/demand of input and output commodities and associated efficiency/conversion to products. These inputs are defined in a few input files:

- 1) a set of geospatial layers defining the network and facility locations
- 2) a scenario configuration file defines key input parameters applied to the network and optimization
- 3) a set of comma-delimited files to define the facility data

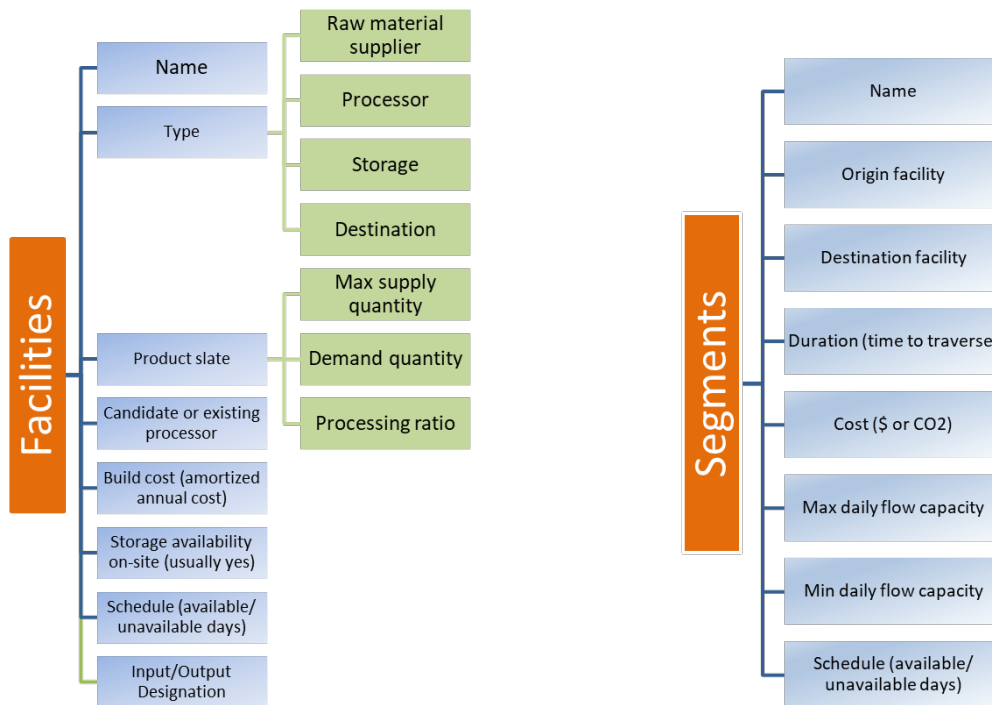


Figure 6: Attributes of the facilities and network segments in FTOT that the user can adjust to build a scenario.

5.1 XML Schema

Key parameters and the location of input data are defined within a scenario configuration file. The scenario configuration file uses the Extensible Markup Language (XML) file format. An XML file is a structured text file that is formatted with markup language making it readable by both humans and machines.

To facilitate the input of many different variables and running of multiple scenarios, an XML-based “scenario file” approach supplies the inputs to the FTOT. This XML Scenario file tags each potential data source as a file source, a function, or a specific value. The XML Scenario file is a text-based file that can be modified in a text editor or integrated development environment for software development. The XML file uses tags to indicate the presences of variables that should be used in a scenario. FTOT can generate a new scenario xml file populated with certain default values for various parameters using the FTOT Tools suite. The user must update the placeholder values in the XML file with scenario specific information, and the default values should be reviewed and adjusted by the user. The XML file is validated against an XML schema file. The ‘Master_FTOT_Schema.xsd’ file is located in the ‘lib’ folder of the project code repository (e.g. C:\FTOT\Program\lib). This schema ensures that the parameters input into the XML are in the format expected by FTOT. If the XML scenario file does not meet the requirements in the schema, FTOT raises an error and reports the offending line and element.

5.2 Configuring FTOT Scenarios in the XML File

All scenario configurations are through the ‘scenario.xml’ file. The FTOT XML specifies input data file locations and user variables for the scenario.

The xml is divided into several sections:

- Scenario
 - Name, description, FTOT version
- Scenario Inputs
 - Base_Network_GDB – location of the multimodal network
 - Base_*_Layer – location of the facility locations for RMP, Destination, and Processors
 - *_Commodity_Data – location of the facility commodity csv files for RMP, Destination, Processors, and Candidate Processors
 - Schedule_Data – (optional) location of the schedule csv file
 - Commodity_Mode_Data – (optional) location of the commodity mode csv file
 - Default_Units_*_Phase – user specified units for the scenario. Mixed units are acceptable in input CSV files. FTOT uses the Pint module in Python to track and convert quantities to the default units specified in the scenario configuration file.
- Assumptions
 - Vehicle load capacity, fuel efficiency, and emissions are specified here. Users are free to modify values as necessary.
- Script Parameters
 - Network_costs
 - Modal costs: railroad, road, barge
 - note: pipeline costs are specified by tariffs values that have been rounded for public release.
 - Intermodal transloading costs
 - Artificial link distances to connect facility to the network

- Route Optimization
 - Network presolve options: (optional) to toggle the network presolve using the NetworkX shortest path algorithm on or off.
 - Default setting if variable is not included is off.
 - Permitted modes: road, rail, water, pipeline_crude, pipeline_prod
 - Allows the user to selectively exclude modes from the optimization analysis
 - Capacity options: to toggle capacity and background flows for the network links on or off.
 - Penalty for not fulfilling depot demand: a penalty used in the optimization to drive flow in the scenario.
- Pulp Solve Options:
 - user specified variables for changing default parameters passed to the optimization solver (max solve time, fractional gap)

The following is a description of each element and parameter:

1. <Scenario xmlns="FTOTv5.0.0"> - The namespace used by the XML schema. This should not be modified by the user. As of FTOT 2020.4.1, the XML schema has not been modified in subsequent versions of FTOT to preserve backward compatibility with scenarios generated with previous versions of the software.
2. <Scenario_Schema_Version> - the version of the schema. Used to verify against changes to the XML schema, and code changes. The convention is: version X.Y.Z, where X = major version of FTOT, Y = breaking change to XML/Schema/Code, Z = non-breaking change to XML/Schema/Code. If a breaking change is made to the schema or code, the user must upgrade their XML file to meet the new schema standard.
3. <Scenario_Name> - a short string used to identify the scenario in the reports and logs. Scenario Names should be descriptive but no more than about ten characters long. Note scenarios names should be differentiated for purposes of analyzing results with the Tableau workbooks.
4. <Scenario_Description> - a string used to describe the scenario. This is printed out in the reports and useful to quickly determine what is different between otherwise similar scenarios. For example if the unmet demand penalty is increased in a sensitivity analysis, or if the road network is shut off in the permitted mode list, the description is a great place to note this. Keep descriptions under 500 characters.
5. <Scenario_Inputs> - contains the following elements:
 - a. <Common_Data_Folder> - contains several files that are required by FTOT to run. For example, the default national multimodal network. The ftot_maps.mxd, and tableau_dashboard.twb are used for generating maps and Tableau dashboards, respectively. The files listed in the common data folder are not modified by FTOT. Instead, FTOT copies a local copy to the scenario folder and modifies that version.
 - b. <Base_Network_Gdb> - a string specifying the full path to the network FTOT will setup and use for the scenario. FTOT will copy the base version to the local scenario. Therefore, a single base network can be used across multiple scenarios without impacting the base version.

- c. <Base_RMP_Layer>, <Base_Destination_Layer>, <Base_Processors_Layer> - are defined by a string specifying the full path to the FC containing the respectively facilities that FTOT will use for routing. Again, FTOT will create a copy of the base FC to the local scenario geodatabase and leave the base version untouched. In the case of the <Base_Processors_Layer>, it should be noted that these are user specified processor locations, and not candidates generated by the tool.
 - d. <RMP_Commodity_Data>, <Destinations_Commodity_Data>, <Processors_Commodity_Data>, <Processors_Candidate_Commodity_Data> - are full paths to the CSV files that are used to define the commodity-facility relationships. The RMPs by definition only have “output” commodities that define the available supply from the facility. Destinations only have “input” commodities that define the total demand of a commodity at the destination. Processors have both “input” and “output” commodities that define the facility size and product-slate/processing ratio. The Processor Candidates commodity data file defines the maximum and minimum facility size, processing size, and cost formula for the amortized capital cost of the facility. NOTE: Processors are currently limited to a single input commodity, but may contain multiple output commodities.
 - e. <Default_Units_Solid_Phase> and <Default_Units_Liquid_Phase> - are used to specified the units FTOT will report the results in. The Pint python module is used to convert the quantities and units specified in the commodity input CSV files into the default units specified in these tags. Pint is able to handle a wide range of units, prefixes, and abbreviations. At run time, FTOT will process the default units with a simple conversion test and raise an error if it cannot use the user specified values because of a typo.
6. <Assumptions> - contains the vehicle load and CO2 emission factors for each mode and phase of matter. <Create_Network_Layer_Script> - Updates each network link with the cost functions defined for each mode and link type by the user. <Network_Costs> - The GIS module assigns costs to each link in the multimodal network based on the transport costs specified in the XML Scenario file. The dollar costs on the GIS network are the dollar amounts required to transport material over each particular link. The costs are defined as USD/kilogallon-mile for liquids, and USD/metric ton-mile for solid materials. The default cost values come from the [US BTS](#) average freight revenue per Ton-Mile. The 2004 data was used because it is the latest complete year reported in the table. The 2004 values were scaled to 2017 using the Producer Price Index. The liquid costs were estimated by multiplying the solid values by 2.87 to convert between thousand gallons and metric tons. The impedances apply a scaling factor to the base dollar cost. For example, FTOT is designed to prefer faster/larger links on the road network. Therefore, a multiplier of 1 is used for interstate highways, versus a multiplier of 1.3 for local roadways, resulting in a price of \$0.00045/gallon-mile for interstate highways and \$0.000585/gallon-mile for local roadways.⁴ <Intermodal_Costs>

⁴ Assuming that a large tanker truck holds 8000 gallons, this corresponds to a cost of \$3.60 to \$4.68 per truck-mile, depending on functional class. Truck travel speed was calculated based on the 2007 estimated peak period speed estimated in the FAF for each link, assuming a minimum floor of 15 mph.

- a. <Railroad> - In Volpe's internal FTOT network, rail impedance categories are based on FRA Rail Density Codes, which are sensitive, non-public data. In the public FTOT network, rail impedance categories are based on publicly available rail ownership data and whether or not the rail segment is part of STRACNET. In the public FTOT network, the final three impedance categories (Rail_Density_Code_2_Weight, Rail_Density_Code_1_Weight and Rail_Density_Code_0_Weight) are NOT utilized. These weights are used to encourage flows on portions of the rail network that see higher freight use (internal network) or part of STRACNET and associated with a Class 1 railroad (public network).
 - b. <Truck> - Weights are used to encourage flows on lower functional class roadways (e.g. interstates and highways). Truck_Interstate includes FAF Function Class 1. Truck_Principal_Arterial includes FAF Function Classes 2 and 3. Truck_Minor_Arterial includes FAF Function Class 4. Truck_Local includes all other FAF Functions Classes (excluding those above). More information on functional classes and the FAF road network is available at the following URL:
https://ops.fhwa.dot.gov/freight/freight_analysis/faf/faf4/netwkdbflow/index.htm
 - c. <Barge> - Water impedances are based on categorizing publicly available USACE freight volumes over the national waterway network. These weights are used to encourage flows on portions of the waterway network that see higher freight use. The default value of 10.0 for waterways with no volume is meant to discourage flows on portions of the waterway network with little to no documented freight use. Water_High_Volume includes waterways with volumes >= 10,000,000 tons. Water_Medium_Volume includes waterways with tonnage volumes < 10,000,000 and >= 1,000,000 tons. Water_Low_Volume includes waterways with tonnage volumes < 1,000,000 tons. Water_No_Volume includes waterways with no tonnage volumes documented. The source for the National Waterway Network and waterway network freight volume data is here: <https://usace.contentdm.oclc.org/digital/collection/p16021coll2/id/1450>
 - d. <Intermodal_Transloading_Costs> - Intermodal costs specify the cost of switching from one mode on the network to another (e.g. rail -> road). Intermodal transfers can only take place as specified intermodal facilities on the network. A cost for solid material and a cost for liquid material transloading is specified by the user.
 - e. <Artificial_Links> - Since facilities are not always located directly on the multimodal network, artificial links must be used to connect facilities to the network. A reasonable default is 5 miles, though longer artificial link distances can be conceivably used, particularly for the pipeline network where there is the least amount of detail. Shorter artificial link distances can be used when the facilities are known to be located directly on or adjacent to the relevant network modes.
7. <Route_Optimization_Script>
 - a. <NDR_On> - The True/False flag determines whether the network presolve step using the NetworkX shortest path algorithm is enabled. The default is for the presolve step to be disabled.

- b. <Permitted_Modes> - The following True/False flags determine whether or not a particular mode should be allowed for routing any flows in the scenario. The default is for all modes to be on.
 - c. <Capacity_Options> - The following True/False flag determines whether network capacity should be considered as a constraint for flowing in the scenario. The default is for capacity to be off.
 - d. <Background_Flows> - The following True/False flags determine whether or not a particular mode should have its existing (background) flows considered. If True, then background flows (e.g. existing movements of freight on the network) will be considered. The default is for all modes to NOT have their background flows considered.
 - e. <Minimum_Capacity_Level> - The following setting (0-1) determines the minimum fraction of capacity that must be available for each network segment in the scenario. Setting a value above 0 allows network segments that are already at capacity due to existing flows, to be traversible in this scenario up to the minimum capacity level.
8. <Pulp_Solve_Options> - Settings used to adjust the optimization parameters.
 <Penalty_For_Not_Fulfilling_Depot_Demand> is also called the unmet demand penalty (UDP). This parameter applies a cost to each unit of demand at a destination that is not fulfilled. It is used to drive flow in the scenario. <Pulp_Solve_Options> contains two additional parameters <MaxSeconds> and <fracGap>. These will be deprecated in future versions of FTOT because they are not supported by the PuLP developer.

5.3 Facility Data Input Files

In addition to the XML scenario file, FTOT requires a set of comma-separated-values (csv) files, which define facility-commodity information (e.g., supply/demand amounts, product slates and conversion efficiency, and minimum/maximum facility size, among other elements).

There are three comma-separated-variable input files available for facility definitions in FTOT. These files contain the facility name, facility type, commodity, quantity, units, and input/output flag for raw material producers (RMPs, which are the supply origin locations in the scenario), processors (which are way points or processing locations in the supply chain), and destinations (locations of demand for the final commodities).

Each facility input file includes the following fields:

- Facility_name; used to match the locations in the scenario gdb. NOTE: the facility_name in the input data CSV file must match the records in the facility_name field of the GIS FC. Facility names should be unique and not duplicated. Care should be taken when using county or city names as these are often repeated across states. Additionally, take care that extra trailing whitespace is not included, as “Whitewater” and “Whitewater ” will not match.
- Facility_type, specifies the type of facility.
- Commodity; a unique name for the commodity. FTOT will create origin-destination pairs by matching the commodity names. For example, if an RMP specified commodity_A as

an output, and a destination specified commodity_A as an input, FTOT would try and flow material from the RMP to the Destination. As noted above, care should be taken to avoid extra whitespace in the commodity names.

- Value; the quantity of the commodity. Use of this field varies by facility type: for RMPs, it is the maximum available material; for destinations, it is the total demand; for processors, it defines how much output is created by the specified quantity of input. If a processor has more than one input commodity listed, it is assumed that all commodities must be used in the specified ratio.
- Units; the units of the quantity for the commodity.
- Phase of matter; solid or liquid, (gas is not currently supported)
- IO; specifies if the commodity is an input (i) into the facility, or an output (o) out of the facility. RMPs by definition ONLY have outputs, and destinations by definition ONLY have inputs. Processors must have at least one input AND at least one output commodity.
- Schedule (optional); a user specifies the name of the availability schedule that each facility follows. The schedule name provided in the facility csv should match one provided in the schedule csv input file. The amount of product supplied or demanded at the facility on each day of the scenario is equal to the product of the commodity quantity and the fraction availability provided in the schedule.
- Max_transport_distance (optional); a user specified maximum transport distance (miles) that this commodity can travel. This distance should be excluded from scenario runs in which all processors are fully defined for performance reasons. **However, note that this field is required in the Raw Material Producers input CSV file when FTOT is generating candidate processors.**
- Max_processor_input (optional); here the user can specify the maximum amount of material a processor can take in. If this column exists, it should be the same for all rows of a single processor, not differ by commodity. If this column does not exist, processor capacity is unbounded.

5.4 Facility Geodatabase (GDB)

Contains facility names and locations of destinations (dest), processors (proc), and raw material producers (rmp). Feature classes (FC) contain facility names and point locations of facilities. **Note: facility names must be unique** e.g. Middlesex County CT must be differentiated from Middlesex County MA. FTOT will match the geospatial location specified in the FC to the facility-commodity relationships specified in the CSV files defined above using the facility_name fields.

5.5 Schedule Input File

Schedules can be used to run scenarios that span multiple days. This optional input file allows the user to define schedules of facility availability for each day in a scenario. An availability of 1 indicates that the facility produces or demands the same amount of the commodity as in the “quantity” column of the facility input file. An availability of 0 indicates the facility does not produce or demand any product on

that day, and an availability of 1.5 indicates the facility produces or demands 1.5 times the amount in the “quantity” column in the facility input file.

The schedule input file includes three columns: schedule, day, and availability. Each schedule’s default value is indicated by day ‘0’ and any days with a different availability must be specified. All schedules for a scenario are the same length. The scenario schedule length is determined by the highest value in the ‘day’ column. FTOT forces all schedules for a scenario to be the same length to avoid mismatch issues between facilities with schedules of different lengths. An example of a schedule input is shown in Table 1.

Table 1. Example of schedule input file containing two 7-day schedules.

schedule	day	availability
weekdays	0	1
weekdays	1	0.5
weekdays	7	0.5
exceptDay3	0	1
exceptDay3	3	0

The table has two schedules. The ‘weekdays’ schedule has an availability of 0.5 on days 1 and 7 and availability of 1 on days 2 through 6. Since the largest value in the day column is 7, that is the scenario schedule length. As a result, the exceptDay3 schedule is also 7 days long even though only the default value, and day 3 value are specified. The ‘exceptDay3’ schedule has an availability of 0 on day 3 and an availability of 1 on days 1-2 and 4-7. An example schedule file is available in Quick Start 2 Exercise 4.

The file name and file path should be added to the scenario xml through the Schedule_Data element, after the Processors_Candidate_Commodity_Data element and before the Commodity_Mode_Data element (if included) or Default_Units element.

5.6 Commodity Mode Input File

This optional input file allows the user to toggle different modes on and off for individual commodities. This csv file can be stored in the same place as the other scenario input data, for example the facility commodity csv files, are stored.

The file name and file path should be added to the Commodity_Mode_Data element in the scenario xml, after the Processors_Candidate_Commodity_Data element and before the Default_Units element:

```
<Commodity_Mode_Data>C:\FTOT\scenarios\YOUR_SCENARIO_FOLDER_HERE\commodity_mode.csv</Commodity_Mode_Data>
```

This csv file must be included in order to include pipelines in the solution. Pipelines are included in the FTOT network, but disabled for all commodities by default. This is to allow users to have flexibility in naming commodities while preventing commodities that are not supposed to flow on pipeline from utilizing it in the optimal scenario since it is usually the least expensive mode.

Several example scenarios that use the commodity_mode.csv are provided in the QS8 Pipelines quick start folder.

Table 2 - commodity_mode.csv for two commodities: crude oil, and a generic petroleum product (pet_prods). Crude oil is only allowed to flow on the crude pipeline network (pipeline_crude), and not allowed in the petroleum products pipeline (pipeline_prod). Similarly, the generic petroleum product (pet_prods, e.g. gasoline, diesel, jet fuel) commodity is not allowed to flow on the pipeline_crude network, but is allowed to flow on the pipeline_prod network.

commodity	road	rail	water	pipeline_crude	pipeline_prod
crude_oil	Y	Y	Y	Y	N
pet_prods	Y	Y	Y	N	Y

5.7 Default Data/Input Parameters

During FTOT development, certain costs, impedances and emissions factors were identified for scenario analyses and testing. These values are included in the FTOT default xml and network as a starting point for new users, but can be modified as desired. A few key default parameters and their sources are described below.

Table 3: Modal cost units and default values in FTOT.

Transport mode	Cost (solids, per metric ton-mile)	Cost (liquids, per kgal-mile)
Roadway/Trucking	0.19	0.54
Railway/Rail car	0.03	0.09
Waterway/Barge	0.02	0.07
Pipeline	N/A	Actual tariff cost (varies by route)
Transloading cost	12.35	40.0

Source for costs—[Bureau of Transportation Statistics Average Freight Revenue per Ton-Mile](#)—2004 data annualized to 2017 using the Producer Price Index

The default impedances are listed in Table 4.

Table 4: Modal impedance factors applied to the base modal cost based on modal link category (road) or existing flow bin (rail, waterway).

Mode	Category / Bin	Impedance Factor
Road		
	interstate	X 1.0
	principal arterials	X 1.1
	minor arterials	X 1.2
	local roads	X 1.3
Rail (public network)		
	STRACNET (Class 1 Ownership)	X 1.0

Mode	Category / Bin	Impedance Factor
	Other STRACNET	X 1.1
	Class 1 Ownership (non-STRACNET)	X 1.2
	Class 1 Rights (non-STRACNET)	X 1.3
	Non-Class 1 (non-STRACNET)	X 1.4
Waterway		
	high volume links (> 10 million annual tons)	X 1.0
	medium volume links (1 – 10 million annual tons)	X 1.3
	low volume links (< 1 million annual tons)	X 1.6
	0 gross tons	X 10.0

Impedances listed in the table above are based on attributes contained within FTOT's source network data. See Section 3.2.1 for more details on how to access these source datasets.

Network Capacity and Existing Flow

The capacity constraint can be enabled by mode to reroute flows once available capacity has been met. Network capacity and background volumes are collated or estimated from existing data sources and are included in the default multimodal network that is supplied with FTOT. Capacity is defined by the number of vehicles or vessels that can pass over a link in a 24 hour period.

- Road
 - Background flows and capacity data come from the Freight Analysis Framework Version 4 (FAF4) dataset
 - FTOT's processed version of the FAF converts average annual daily traffic (AADT) to unidirectional daily flows
 - FTOT's processed version of the FAF converts hourly capacity to daily unidirectional capacity
 - Available capacity = 1- (daily traffic / daily capacity)
- Rail
 - For Volpe's internal network, background flows are based on FRA's NARN, which includes segment-level estimated freight tonnage flows (NOTE: These background flow tonnages are non-public, sensitive data and neither the tonnages nor their binned counterparts are included in the public FTOT network).
 - The public FTOT network does not contain any background flow data for rail, as only publicly available attributes from FRA'S NARN are included in FTOT's rail network.
 - For both the internal and public networks, daily unidirectional carload capacity is estimated based on track number and signaling type publicly available in NARN (and based on methodology developed by [Cambridge Systematics National Freight Infrastructure Capacity & Investment Study 2007](#))
- Waterway

- FTOT assumes waterway capacity is only an issue at typical waterway chokepoints (e.g. locks)
 - Background flow and capacity data is based on US Army Corps Lock Characteristics and Lock Report data.
 - FTOT network uses this data to estimate lock-specific barge equivalent flows and lock-specific barge equivalent capacity.
- Pipeline
 - For crude oil pipelines, Oak Ridge National Laboratory's TRIM model estimates existing flows and capacity data for crude oil pipelines. This data is integrated into the default FTOT network. Crude pipelines not in TRIM are left without capacity constraint.
 - A source for product pipeline background flow and capacity data has not yet been identified, so these pipelines remain without capacity constraint in FTOT.

6 FTOT OUTPUTS

FTOT generates four types of outputs. These are:

- 1) A human-readable text report containing the configuration history, results, elapsed run time, and any warnings given during the scenario.
- 2) A CSV file report that can be used to generate graphical dashboards showing summary map and statistics for the scenario, such as relative contributions of commodities and mode to cost, emissions, and VMT, amount of supply utilized and demand met in the scenario, and other elements. The CSV file can be used as an input to Excel, Tableau, or other data analysis programs.
- 3) A package tableau workbook (.twbx) that contains all the geospatial and summary result information required to display a Tableau dashboard in the free version of the Tableau Reader software.
- 4) Maps from each step of the analysis, including the network, facilities, candidate processing locations, and the optimal solution (optimal facilities, optimal flows, non-optimal facilities, etc.).

In addition to the four outputs listed above, the scenario results are stored in a sqlite database and in the scenario geodatabase. The user is free to create their own analysis or tools for reviewing this data.

6.1 Reports

Report outputs are generated as a csv file that is stored in the scenario folder. The csv report includes three groups of reporting results: the scenario configuration, the commodity-specific summary results and the facility summary results.

The scenario configuration details, reporting out the scenario name, description, and various input parameters from the xml, as well as the name and location of the GIS layers used to define the network and the facilities included in the scenario.

The commodity summary results include the following data by commodity and mode:

- Mass and volume movements
- Cost of movements
- Fuel burn
- Miles
- Vehicles
- VMT
- CO2

The facility summary results include

- Facility utilization (amount of supply utilized or demand met)

- Facility utilization fraction (relative to total supply or demand)

6.2 Graphics Dashboard

A graphical dashboard output is created using Tableau (see example figures in accompanying Quick Start Documentation).

The Tableau dashboard is used to visualize the CSV report, and GIS facilities and routes from the analysis. It includes facility-level, commodity-level, and mode-level results. The results include quantities of material moved, utilization levels, CO₂, VMT, mileage, fuel burn, and costs.

To view the dashboard in Tableau:

- Navigate to the reports sub-directory in the scenario folder, and select the latest tableau report folder. Double click on the tableau_dashboard.twbx file to open it in Tableau Reader.
- On the dashboard there are nine charts (NOTE: Charts are grouped by commodity type, and sorted alphabetically by mode).
 - From left-to-right, top-to-bottom:
 1. Key Input Parameters table
 2. Optimal Solution Routes by Mode (map)
 3. Size & Utilization of Facilities (map)
 4. Total vs. Optimal Supply & Demand Quantity
 5. Material Moved by Commodity & Mode
 6. Scenario Cost by Commodity & Mode
 7. Total vs. Optimal Supply & Demand Facilities
 8. VMT by Commodity & Mode
 9. CO₂ Emissions by Commodity & Mode
- Notes & Tips on Dashboard Symbolology
 - Mode colors:
 - Pipe – brown
 - Rail – purple
 - Road – red
 - Water – blue
 - Facility Type colors:
 - RMP Optimal Supply – dark green
 - RMP Total Supply – light green
 - Destination Optimal Demand – dark orange
 - Destination Total Demand – light orange
- Notes & Tips on Dashboard Functionality
 - In upper-right corner, the dropdown menu changes what facilities are viewed on the map “Size & Utilization of Facilities” map. The default view is “Demand”.

- Hover over an item on a chart (e.g. bar, icon, point, etc.) to display tooltip which contains additional data.

6.2.1 Tableau Scenario Comparison Dashboard

Released in 2020.1, the Scenario Compare tool is a [supplemental FTOT Tool used to](#) concatenate the results from one or more scenarios in a list.

The tool has two modes to generate the list of scenarios to concatenate: an automatic recursive search, and a user-specified search. In the recursive mode, the user provides a high level directory and FTOT returns all sub-directories within it. In the manual mode, the user specifies which scenario folders should be added to the list until specifying the “done” keyword.

After the list is established (either automatically with the recursive search, or manually) the tool will search for tableau workbooks within the .\Reports subfolder, find the latest version, and concatenate the tableau_report.csv and tableau_output.gdb.zip records into the compare files. Any subfolders without a Reports folder are skipped.

Once the data is concatenated, the tool zips the workbook, csv report, geodatabase, and supplement files into a packaged workbook file (.TWBX). The user can open this packaged workbook using Tableau Reader, which is available as a free download. The comparison dashboard allows the user to select one or more scenarios to compare against.

The 2020.2 release added the ability to compare multiple scenarios in the Tableau dashboard. New functionality includes options to change the color of mapping route results by mode, commodity, and scenario name, as well as a Tableau Story format that allows users to step through each of the following dashboards:

- Supply Chain Summary
- Route Results by Commodity and Mode
- Facility Results by Supply and Demand
- Run Times
- Parameters

6.3 Maps

The FTOT map outputs are generated in the scenario’s Maps directory (see example figures in accompanying Quick Start Documentation) and include the following:

- The multimodal network used in the optimization (01_S_Step)
- Raw Material Producers (02a_F_Step_RMP, 02a_F_Step_With_Labels)
- User Defined Processors (02b_F_Step_User_Defined_PROC, 02b_F_STEP_User_Defined_PROC_With_Labels)
- Ultimate Destinations (02c_F_Step_DEST, 02c_F_Step_DEST_With_Labels)
- All Facilities (02d_F_Step, 02d_F_Step_With_Labels)

- Processor Candidates (03a_F2_Step_Processor_Candidates, 03a_F2_Step_Processor_Candidates_With_Labels)
- All Processors (03b_F2_Step_Processors_All, 03b_F2_Step_Processors_All_With_Labels)
- Final Optimal Routes (04a_O_Step_Final_Optimal_Routes_with_Commodity_Flow, 04b_O_Step_Final_Optimal_Routes_with_Commodity_Flow_NO_LABELS, 04c_O_Step_Final_Optimal_Routes_with_Commodity_Flow_JUST_FLOW)
- Optimal and Non-Optimal Raw Material Producers (04d_O_Step_Optimal_and_Non_Optimal_RMP)
- Optimal and Non-Optimal Processors (04e_O_Step_Optimal_and_Non_Optimal_PROC)
- Optimal and Non-Optimal Ultimate Destinations (04f_O_Step_Optimal_and_Non_Optimal_DEST)

All map names are suffixed with the following text-- “default_basemap”, “gray_basemap” or “topo_basemap”-- based on the version of the mapping that is specified in the run.bat file for the scenario.

Running the default “m” step produces maps utilizing the default FTOT basemap, replacing “m” with “mb” produces maps utilizing the Esri light-gray basemap, and replacing “m” with “mc” produces maps utilizing the Esri topographic basemap. In general, the default basemap is adequate for national and large regional scenarios, while the gray and topographic basemaps are more appropriate for smaller regional and local scenarios.

Note that many maps include variations that include/exclude labels. Since each scenario is different and all the maps are automated, some variations may work better than others depending on the scenario.

The user is welcome to manually develop their own maps within ArcGIS leveraging the map document automatically generated in each scenario’s output. If a user wishes to produce customized maps that are automatically output directly to the maps folder within an FTOT scenario, they must edit the ftot_maps.mxd provided in the FTOT common_data folder and add user-specified and user-symbolized map data within the group layers located under the “CUSTOM_USER_CREATED_MAPS” section of the map document’s Table of Contents. If these group layers are populated with actual data, FTOT will export these maps when the mapping step is run. If the optional m2 Step of FTOT is run, additional time and commodity specific mapping is output into the scenario’s Maps_Time_Commodity folder. These include:

- Maps showing location of flows for each commodity in the scenario
- Maps showing location of flows for each time step in the scenario
- Maps showing location of flows for each commodity/time step combination in the scenario
- An animation (.gif) representing flows for all time steps in the scenario

Like the m step, alternative basemaps can be run by replacing the m2 step in the bat file with m2b (gray basemap) and m2c (topographic basemap).

7 SETTING UP A SCENARIO

As previously discussed, FTOT requires several input files to fully specify a scenario.

A scenario configuration file (scenario.xml) is used to define the location of key input data such as the multi-modal network, facility locations, and facility commodity information defined by the user, as well as key input parameters and assumptions for the run.

A batch file (run.bat) is used to automatically run the sequence of steps necessary for a complete FTOT scenario. The batch file references the scenario configuration file, which is loaded at runtime during each step. The file also references the Python path and FTOT directory.

Both of these files can be generated by the FTOT Tools suite.

Alternatively, the Quick Start scenarios and documentation are designed to familiarize the user with several scenarios of various complexities and numerous exercises for the user to explore. The quick start scenarios can also be used as a template for setting up new scenarios.

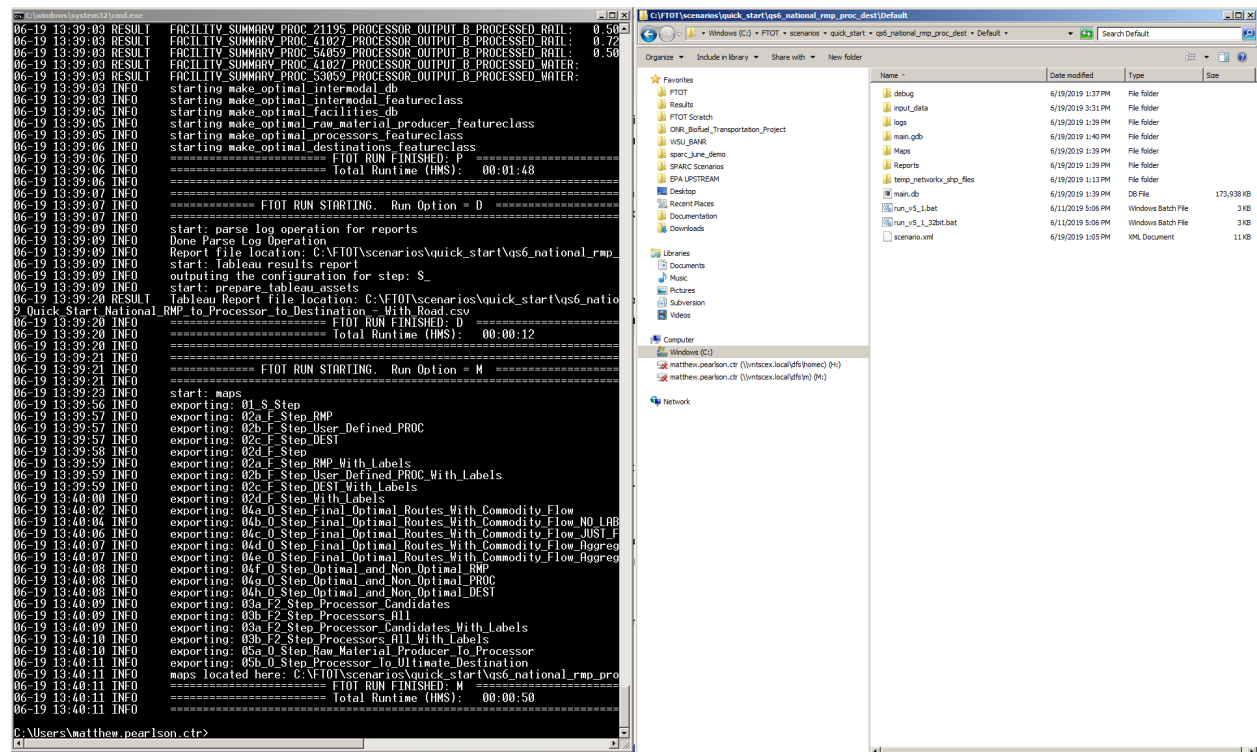


Figure 7 - Split screen practice for running FTOT scenario. The command line is on the left, and the scenario opened in the Windows Explorer is on the right.

8 RUNNING A SCENARIO

FTOT is a command line program that executes in the Windows shell. It is a recommended practice to run the scenarios with a command line window on the left, and the scenario folder opened in an explorer window on the right. This will allow the user to see that the scenario running in the appropriate folder.

A scenario can be started by dragging the run.bat file located in the scenario folder into the command line and hitting the [enter] key. During the run, information is logged to the command line providing progress information and results for each step. Detailed log files are created for each step and timestamped in the .\logs folder. As new files are created in the various FTOT steps, they will become visible in the explorer window on the right.

9 SUPPLEMENTARY TOOLS

A set of supplementary FTOT tools is provided with the program to assist the user with a variety of common tasks. These tools include the following:

- **xml_tool:** Generates an xml scenario file based on either a 1) FTOT template with defaults or 2) existing, older user-generated xml file.
- **bat_tool:** Generates a bat file for running a new FTOT scenario based on user-provided inputs.
- **scenario_compare_tool:** Concatenates the results from one or more scenarios into a packaged Tableau workbook. The tool has two steps: (i) the user specifies an output directory where the concatenated results workbook will be stored, (ii) the user specifies the input directories which contain the results to be concatenated. This can be done recursively by supplying a top level directory to automatically generate a list of sub-directories, or the user providing the scenario directories individually.
- **aggregate_raster_data:** coming soon.
- **generate_template_csv_files:** Generates the input data csv files for RMPs, processors, and destinations. The template for RMP input does not include the optional max_transport_distance field, so the user should add this if necessary (notably when running candidate generation scenarios).
- **breakpoint:** sets a python debugger breakpoint and enters a debugging REPL
- **replace_xml_text:** a tool to batch replace XML configuration elements recursively through a top-level directory. Useful for making changes to a number of scenario files, such as changing the base network gdb location.

The FTOT tools interface can be accessed by running the following in a command-line window:

32-Bit Python

“You will need ArcGIS 10.1 or later to run this script. Exiting.”

Assuming the user does indeed have ArcGIS installed, confirm that the license to use ArcGIS is available (try opening up ArcGIS in its own window). If not, you may need to connect to your work network or a VPN in order to access the license—contact your ArcGIS license administrator for further details.

- Import errors. Python is missing a required module. See the installation guide [<https://github.com/VolpeUSDOT/FTOT-Public/wiki>], and verify that the module is installed in the Arcgis python environment.

- If a user is having difficulty running a batch file (run.bat) or process, the user may need to include ‘python’ on the lines specifying the process. For example, the user would insert a ‘python’ before the syntax that runs the process. In pseudocode:

```
python PythonScriptFilePath XMLfilePath ProcessYouWantToRun  
USAGE EXAMPLE: python C:\FTOT\Program\python.py  
C:\FTOT\Scenarios\midwest1\scenario.xml
```

- If the user has a prior version of Python installed, there may be some conflicts with the new Python install included with ArcGIS. In that case, the user may need to add the following line to the BAT file:

```
path C:\Python27\ArcGIS10.6;C:\Program Files (x86)\ArcGIS\Desktop10.6\bin;%path%
```

- If the user is unable to install the ArcGIS 64-bit background geoprocessing, the user must run FTOT in 32-bit.
 - For Quick Start scenarios, run the batch files ending in run_32bit.bat.
 - For large or complex scenarios (e.g. Quick Start 6, a national scenario), FTOT may not run properly in 32-bit.
- If FTOT cannot find the GIS feature classes or facility commodity CSV files it will throw an exception error and exit the program. If this happens ensure that the appropriate scenario assets exists and the path is properly specified in the XML.
- On some machines, FTOT may fail during the O2 step with an “IOError: [Errno 13] Permission denied”. This may be due to specific constraints within your work network or VPN. Try running FTOT on a different network and/or off the VPN to see if it resolves the issue.
- Possible causes of no-flow solutions:
 - The unmet demand penalty is too low. Adjust the unmet demand penalty in the scenario configuration file. In general, it may be necessary to raise this penalty when any other cost (e.g. rail transport) is raised, or else the optimizer will conclude that it is

more optimal to transport less material. As a general guide, the unmet demand penalty will likely work best if set to be 10-50 times the average actual transportation cost. It is advised to run scenarios with multiple unmet demand penalties to explore the sensitivity of a given analysis.

- The candidate processing facility capital cost is too high. Increase the unmet demand penalty in the scenario configuration file, or decrease the capital cost in the processors candidate commodity CSV file.
- The processing facility lower bound are too high. Adjust the minimum processor size in the facility commodity CSV file. In scenarios involving candidate generation, a no-flow solution will occur if a value for maximum transport distance is not specified in the raw material producer commodity CSV file. Ensure that a maximum transport distance (in miles) is specified.
- No-flow solutions are possible if facilities are not located close enough to segments in the FTOT transportation network. To address, raise the artificial link distances specified in the scenario's XML to ensure facilities connect to the network (for most modes, the default artificial link distance is 5 miles). Lack of connectivity is unlikely unless facilities are located far away from populated areas or are located outside of the continental United States, which is the current extent of the FTOT transportation network.
- For capacity-constrained scenarios where existing background flows are being considered, no-flow solutions are possible when facilities are not accessible due to capacity constraints. Turn off the capacity constraint or increase the minimum available capacity in the scenario's XML to address this issue.

10.1 Known Bugs and Issues

- Errors with very large numbers:
 - If it seems like numbers are being capped or cut off in unexpected ways, the most likely cause is an improperly set constraint – double check the upper and lower bounds on processing facility flow, transportation and transloading costs, and demand vs production values. If none of those address the problem, it is possible that limitations of the programming language in handling very large numbers are having an impact. The COINMP_DLL solver is written in C++, which limits values depending on the data type to $\pm 2,147,483,647$ (Integer), 38 places (Float), 308 places (Double).
- In versions of ArcGIS below ArcGIS 10.6 (which are not fully supported by FTOT), the ArcGIS map document (mxd) file accompanying FTOT and exported in scenario results will not open properly. To access this mxd and the ability to create custom scenario maps in ArcGIS, the solution is to upgrade to ArcGIS 10.6 or higher. However, on earlier versions of ArcGIS FTOT should still run end-to-end and all default scenario maps and reports will still be generated.

11 ACKNOWLEDGEMENTS

This research was funded by the U.S. Federal Aviation Administration Office of Environment and Energy through FAA contract 693KA9-18-N-00022 under the supervision of Nathan Brown. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the FAA.

12 REFERENCES

- "pulp-or". 2007. COIN-OR.
- Bertsimas, D., and J. N. Tsitsiklis. 1997. Introduction to Linear Optimization. Athema Scientific, Belmont, MA.
- Bureau of Transportation Statistics. 2011. Intermodal Terminal Facilities, available at http://www.rita.dot.gov/bts/sites/rita.dot.gov.bts/files/publications/national_transportation_atlas_database/2011/index.html. National Transportation Atlas Database (NTAD)
- Carreira-Perpiñán, M. Á. 2014. Simplex Method. MathWorld--A Wolfram Web Resource, created by Eric W. Weisstein.
- COIN-OR. 2007. Clp.
- Federal Highway Administration. 2013. Freight Analysis Framework (FAF) version 3.4.
- Leonard B. Levine Associates. 2014. US Petroleum Product Pipeline Movements OD Pair and Base Rate Report. len.levine@lensonwashington.com.
- McDuffie, W. 2013. Teleconference discussion of fuel transport costs. Volpe National Transportation Systems Center.
- pulp-or. 2014. puLP: An LP modeler in Python.
- Surface Transportation Board. 2011. Carload Waybill Sample.
- United States Army Corps of Engineers, and Bureau of Transportation Statistics. 2011. U.S. Army Corps of Engineers Navigable Waterway Network, available at http://www.rita.dot.gov/bts/sites/rita.dot.gov.bts/files/publications/national_transportation_atlas_database/2011/index.html. National Transportation Atlas Database (NTAD)
- USDA. 2014. National Agricultural Statistics Service: Quick Stats.