USER'S GUIDE AND MODEL DOCUMENTATION: AGENT-BASED SUPPLY CHAIN MODELING TOOL

the science of insight

5.6.2015

PREPARED FOR:
CHICAGO METROPOLITAN AGENCY FOR PLANNING

SUBMITTED BY:
RSG

55 Railroad Row
White River Junction, VT 05001
802.295.4999
www.rsginc.com

IN COOPERATION WITH:
DAVID WU, LLC, SUNIL CHOPRA, MATTHEW ROORDA

USER'S GUIDE AND MODEL DOCUMENTATION: AGENT-BASED SUPPLY CHAIN MODELING TOOL

PREPARED FOR:
CHICAGO METROPOLITAN AGENCY FOR PLANNING

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0 OVERVIEW

The purpose of this document is to assist the user in setting up and running the Chicago Regional Meso-Scale Freight Forecasting Model with Agent-Based Supply Chain Evolution, including integrated runs with the Chicago Regional Truck Touring Model. The models were developed by RSG for the Chicago Metropolitan Agency for Planning (CMAP). This document includes the following sections:

- Hardware and Software Requirements
- Model Design
- Model Database
- Developing Alternative Scenarios
- Appendices Containing Additional Software Documentation

1.1 HARDWARE AND SOFTWARE REQUIREMENTS

The application program package is the software that implements a set of analytical modeling components and system-level algorithms that simulate the evolution of globally-connected supply chain relationships in the Chicago region and how these trading relationships translate to regional freight flows defined by industry, commodity, size, and mode. The core application program is called rFreight™ and is written in the open-source R statistical programming language. The core program executes an additional binary file that performs the more computationally intensive agent-based simulation. This section of the user's guide covers three main topics:

- **System Requirements** – The Chicago Regional Meso-Scale Freight Forecasting Model with Agent-Based Supply Chain Evolution has greater computational requirements than the first generation of the Chicago Regional Meso-Scale Freight Forecasting Model. This section describes the memory and hard-disk space requirements as well as how to configure the model for distributed processing on a single multi-core machine.
- **Meso-scale Supply Chain Model** – This section describes the R scripts and R packages used in the meso-scale supply chain model. It includes all of the data input and output processing and the analytical modeling steps, and calls the binary file for the agent-based simulation.
- **Procurement Market Game (PMG)** – This section describes the software aspects of the PMG program, which operates as a processing kernel within the larger meso-scale freight model (rFreight™) to simulate the evolution of supply chain relationships. The installation package includes an executable file and C++ source code.

1.2 MODEL DESIGN

The Meso-Scale Freight Forecasting Model with Agent-Based Supply Chain Evolution differs from previous versions of the regional freight-forecasting tool. The most obvious change, the addition of the agent-based supply chain evolution module, extended the previous meso-scale model; in doing so, it brought about changes to the ordering of some of the other analytical modeling components.

1.3 MODEL DATABASE

This section describes the "Base" scenario database, which contains both input and output sections. The set of inputs consists of three types of files: (1) model files, which contain parameters for individual model components or calibration target values; (2) data files, which contain lookup tables on costs, production and consumption rates, and skims and zone information; and (3) industry and commodity-code correspondence files. The output section contains a series of commodity-specific output files that record the number and quantities of transactions.

1.4 DEVELOPING ALTERNATIVE SCENARIOS

The Meso-Scale Freight Forecasting Model with Agent-Based Supply Chain Evolution is fundamentally a scenario-modeling tool. This section describes how to configure model inputs and run-time parameters to perform certain types of analysis.

2.0 HARDWARE AND SOFTWARE REQUIREMENTS

The Chicago Regional Meso-Scale Freight Forecasting Model with Agent-Based Supply Chain Evolution is a computationally intensive agent-based simulation model; thus, it has several hardware and software requirements for a model user to run the model successfully. In addition, there are several additional software requirements for the model user to be able to compile the source code of various elements of the model when updating source code. This section describes the systems requirements for running the model, the software elements of the meso-scale supply chain model, and the software required to compile and use the PMG application.

2.1 SYSTEM REQUIREMENTS

The model has greater computational requirements than the first generation of the Chicago Regional Meso-Scale Freight Forecasting Model. This section describes the memory and hard-disk space requirements as well as how to configure the model for distributed processing on a single multi-core machine. The current version of the model is not configured to run in a distributed manner across multiple workstations.

2.1.1 OPERATING SYSTEM

The model has been developed and tested on PCs running Windows 64-bit operating systems, including Windows Server 2008 R2 Enterprise, Windows 7 Ultimate, and Windows 8.1 Pro. Given the memory requirements discussed below, the operating system must be a 64-bit operating system in order for a 64-bit version of R to be run and utilize a sufficient amount of memory.

2.1.2 HARD-DISK REQUIREMENTS

The hard-disk space requirements for installing and running the model are primarily governed by the space needed for scenario outputs, particularly if multiple separate scenarios are run. The base installation is approximately 200 MB and includes base scenario inputs and all model source and compiled code.

The outputs from a scenario amount to approximately 100GB if all of the commodity markets are run, although this amount varies considerably depending on the settings used in the model run for logging (e.g., creating verbose log files from each PMG run). Running multiple scenarios adds disk space requirements (in approximately 100GB increments) if the complete outputs from each run are retained, particularly if the separate R workspace and .csv output for each individual PMG are saved, since these files consume a large proportion of the disk space used for model outputs.

2.1.3 RAM REQUIREMENTS AND USAGE

The model runs in two modes. In the early and late components of the model, R runs a single processing thread with several relatively large tables manipulated in memory. During these operations, the minimum available RAM requirement is approximately 30-40GB. Note that on any system, the RAM available to run the model does not equal the installed RAM—as some RAM is always in use by the operating system and any other applications that are running, including the many background processes that run on Windows machines.

The second run mode happens during the middle portion of the model, where multiple, separate Rscript.exe processes run individual commodity markets and execute the PMG application. This distributed processing mode does not necessarily require more RAM than the 30-40GB mentioned above, but if additional RAM is available the model run time can be reduced. The setting to achieve this and the resulting RAM requirements are discussed in the next section.

2.1.4 DISTRIBUTED PROCESSING

Distributed processing involves splitting a process into multiple separate calculations, executing those calculations on separate processors or threads, and then combining the results back together. As previously mentioned, the model follows this approach by running each individual commodity market in a separate process, where Rscript.exe is used to run the model and call the PMG application at the appropriate time to run the PMGs for that commodity market.

The model user has control over how many individual processes are run and should set the *maxrscriptinstances* variable in the scenario_variables.R script for the scenario (see Section 0) so that the available RAM on the machine is not exceeded. The model creates one monitoring process and then *maxrscriptinstances* -1 processes to run the PMGs (which means the minimum value of *maxrscriptinstances* is 2).

The size of a single process to run the PMGs is dependent mainly on the number of producers and consumer combinations used; the size of the data files manipulated in memory is largely a function of the number of rows in the costs files, which includes a row for each possible producer and consumer pairing. The base value of the variable *combinationthreshold* (also set in the scenario_variables.R script) is 7,000,000, which leads to each process running the PMGs using up to approximately 4GB of RAM.

Therefore, on a computer that had 48GB of RAM available, setting *maxrscriptinstances* = 12 is appropriate: the 11 processes that run PMGs will use up to around 44GB of RAM and the combination of the monitoring process and the main R process will use some portion of the remaining 4GB of available RAM. This leads to a rule of thumb of setting *maxrscriptinstances* = available RAM/4 (if the base value of *combinationthreshold* is used). However, on a given computer system some experimentation is warranted, as using high proportions of the resources (RAM, processors) can result in slower run times than slightly lower proportions.

2.2 MESO-SCALE SUPPLY CHAIN MODEL

This section describes the software required to execute the R scripts that comprise the meso-scale supply chain model and the additional R packages used during a model run. The meso-scale supply chain model includes all of the data input and output processing and the analytical modeling steps, and calls the PMG application for the agent-based simulation.

2.2.1 R AND R STUDIO

The model requires two open-source and freely available software applications in order to run: R and R Studio, an integrated development environment for R. Both those software are required to run the model from R Studio; currently the model does not connect and disconnect from the PMG application correctly when using RGui, the built-in graphical user interface which comes packaged with R.

- The latest version of R for Windows is available from http://cran.r-project.org/bin/windows/base/. The latest version (at the time of publication of this document) is R 3.1.3 (http://cran.r-project.org/bin/windows/base/R-3.1.3-win.exe). Installation instructions are at http://cran.r-project.org/bin/windows/base/README.R-3.1.3

- The latest version of R Studio for Windows is available from http://www.rstudio.com/products/rstudio/download/. The latest version (at the time of publication of this document) is RStudio Desktop 0.98.1103 (http://download1.rstudio.org/RStudio-0.98.1103.exe).

2.2.2 STANDARD R PACKAGES

In addition to the base functionality contained within R, the model uses several R packages, which are additional collections of functions loaded by R at run time. If these packages are not present on the model user's machine at run time, the model will attempt to download and install them (requiring access to the internet and permissions on the machine set so that R can connect to the internet and install files). The packages used by the model are as follows:

- data.table (version 1.9.4): This package speeds up the handling of large tables in the model. The package provides, "fast aggregation of large data (e.g. 100GB in RAM), fast ordered joins, fast add/modify/delete of columns by group using no copies at all, list columns and a fast file reader (fread)" (http://cran.r-project.org/web/packages/data.table/index.html).
- bit64 (version 0.9-4): This package is used to handle large 64-bit integers (http://cran.r-project.org/web/packages/bit64/index.html).
- reshape (version 0.8.5): This package is used to restructure and aggregate data from long to wide format and vice versa (http://cran.r-project.org/web/packages/reshape/index.html).
- reshape2 (version 1.4.1): This package adds additional data restructuring capabilities to those found in the reshape package (http://cran.r-project.org/web/packages/reshape2/index.html).
- ggplot2 (version 1.0.0): This package provides charting capabilities beyond those found in the base R charting functions (http://cran.r-project.org/web/packages/ggplot2/index.html).
- fastcluster (version 1.1.16): This package is used in the truck-touring model to implement the hierarchical clustering used in the formation of tours (http://cran.r-project.org/web/packages/fastcluster/index.html).

In addition to the above R packages used by the model during a run, there are two other packages used to build the rFreight package that can be installed and then loaded in the 0_Create_rFreight.R script:

- devtools (version 1.7.0): This package contains a set of functions and tools that facilitate R package development (http://cran.r-project.org/web/packages/devtools/index.html).
- roxygen2 (4.1.0): This package converts the in-source code documentation included in package source code files into R documentation files, ready for compiling into an R package (http://cran.r-project.org/web/packages/roxygen2/index.html)..

2.2.3 rFREIGHT R PACKAGE

Many of the functions written to execute the model have been combined together into an R package that is included as part of the model. The package is called rFreight. This section of the user's guide introduces the rFreight R package, and explains both how the package is used in the model and how it is compiled from the source files. The documentation for the rFreight package is included as Appendix A to this user's guide.

The model structure, described in Section 2.0, is composed of folders of data and R scripts, and includes the R Freight package zip file (rFreight_0.1.zip) in the root directory of the model. When the model is run, one of the first steps it carries out is to install the rFreight package from the zip file and then load it. The rFreight package includes several functions that deal with various elements of running the model:

- Model management – for example, the *loadInputs* function, which loads a set of files to the global environment.
- Model application – for example, the *predict_logit* function, which supports simulation of logit models.
- Working with freight data – for example, the *naics6naics2* function, which supports aggregation from six-digit North American Industry Classification System (NAICS) codes to top level 2-digit NAICS codes.
- Working with the PMG application – for example, the *writePMGini* function, which writes a complete configuration .ini file for the PMG application given a set of variables.
- Truck-touring – for example, the *sequence_trips* function, which uses a greedy algorithm to sequence the trips that form a truck tour.

In addition to the compiled zip file of the rFreight package, the model includes the source files for the package so the user can modify the functions included in the rFreight package and these changes will be incorporated into the package and recompiled. The package source is included in the rFreight directory, which contains the regular structure of an R package source (e.g., man and R directories, and DESCRIPTION and NAMESPACE files). The rFreight/R folder contains the R script files containing the functions, along with documentation written using roxygen2 style commenting.

If a change is made to a constituent function from the rFreight package by updating one of the files in the rFreight/R folder, the package can be recompiled using scripts/0_Create_rFreight.R script. This script loads the devtools and roxygen2 packages and uses functions from those packages to update the package documentation and then check and build the rFreight package, resulting in an updated rFreight_0.1.zip.

2.3 PROCUREMENT MARKET GAME (PMG)

This section describes the software aspects of the PMG program, which operates as a processing kernel within the larger meso-scale freight model to simulate the evolution of supply chain relationships. The installation package includes an executable file and C++ source code.

The PMG distribution binary was built on Windows using Microsoft Visual Studio Express 2013 for Windows Desktop, although any recent version of Visual Studio should work equally well.

PMG uses the C++ Boost library. Only the tokenizer, format, and algorithm/string libraries are required, so the Header-only version of Boost library is sufficient. The distribution binary of PMG was compiled against version 1.55.0, but no compatibility issues are anticipated if used with eventual subsequent releases. Instructions for installing the Boost library on Windows can be found on the boost.org web site: http://www.boost.org/doc/libs/1_55_0/more/getting_started/windows.html.

The C/C++ source files were compiled for the x64 platform with the following command line options:

/GS /GL /W3 /Gy /Zc:wchar_t /I"C:\Users\user.name\Documents\Visual Studio 2013\boost\boost_85869" /Zi /Gm- /O2 /sdl /Fd"x64\Release\vc120.pdb" /fp:precise /D "_CRT_SECURE_NO_WARNINGS" /D "WIN32" /D "NDEBUG" /D "_CONSOLE" /D "_LIB" /D "_UNICODE" /D "UNICODE" /errorReport:prompt /WX- /Zc:forScope /Gd /Oi /MD /Fa"x64\Release\" /EHsc /nologo /Fo"x64\Release\" /Fp"x64\Release\pmg.pch"

The "include" path to the Boost libraries will need to reflect the corresponding installation location.

The executable is distributed with the msvcp120 and msvcr120 dynamic link libraries (DLLs). These DLLs will not be required to run the executable on a development machine, but will be required to distribute the executable to a machine that does not have Visual Studio installed. On a development machine, copies of these DLLs can be typically in the Visual Studio install sub-directory VC\redist\x64\Microsoft.VC120.CRT.

3.0 MODEL DESIGN

The Meso-Scale Freight Forecasting Model with Agent-Based Supply Chain Evolution extends the previous version of the Meso-Scale Freight Forecasting model through a redesign that allows for the evolution of supply chain relationships through production-sourcing market simulation. The new design no longer pins commodity flows to static inputs from the Federal Highway Administration's (FHWA) Freight Analysis Framework (FAF). Instead, it derives commodity demand endogenously from a base (input) level of consumption and the production requirements for each synthetic firm to meet their production quotas. An agent-based simulation matches consumers and producers of commodities, accounting for tradeoffs in cost, supplier responsiveness, risk, and other factors.

3.1 OVERVIEW

The model starts by synthesizing U.S. firms by industry classification and size, locating them spatially, and deriving annual production and consumption requirements from existing commodity-flow relationships between producing and consuming sectors of the national economy, as represented in U.S. Bureau of Economic Analysis (BEA) input-output (IO) accounts. The model system also synthesizes agents representing firms in countries and industries that currently trade with the United States.

As shown in Figure 1, at the heart of the model system are a set of procurement market models in which firms attempt to fulfill their production needs by purchasing input commodities from other firms that sell the commodity they need. For each commodity market, an iterative PMG is played in which a pool of buyers attempt to procure inputs from a pool of sellers in the market. As an input to the game, the transport-logistics chain models, described in more detail below, simulate the choice of distribution channels, shipment sizes and modes for each prospective buyer-supplier pair, thereby enabling the calculation of logistics costs and shipping times.

Buyers consider shipping times, unit costs (transport and non-transport), and risk minimization (e.g., supply chain disruption); while sellers, who are capacity constrained, evaluate whether to trade with a particular buyer in the face of other, potentially more lucrative offers. Through repeated bi-lateral games, agents form preferences for specific trading partners based on past interactions and may adjust their tolerances for risk based on market constraints.

The final round of the game indicates which agents established trading relationships and the quantities of commodities bought and sold, producing a set of spatially distributed freight flows between firms located in freight analysis zones. In most markets, buyers will far outnumber sellers; however, buyers will likely purchase commodities from multiple sellers, for reasons of risk minimization or limited individual seller capacity. Because foreign buyers and sellers are included in the procurement market, the model also predicts import and export flows, which are limited to observed levels of U.S. imports and exports at an aggregate level.

The outcome of the PMGs are contracts for commodity purchases between buying and selling firms, expressed in annual pounds of goods shipped, average shipment sizes and modes to be used. The model converts these commodity flows into daily shipments by the modes specified in the contract. Truck-borne shipments are converted into vehicle loads and then trip tables, ready for network assignment.

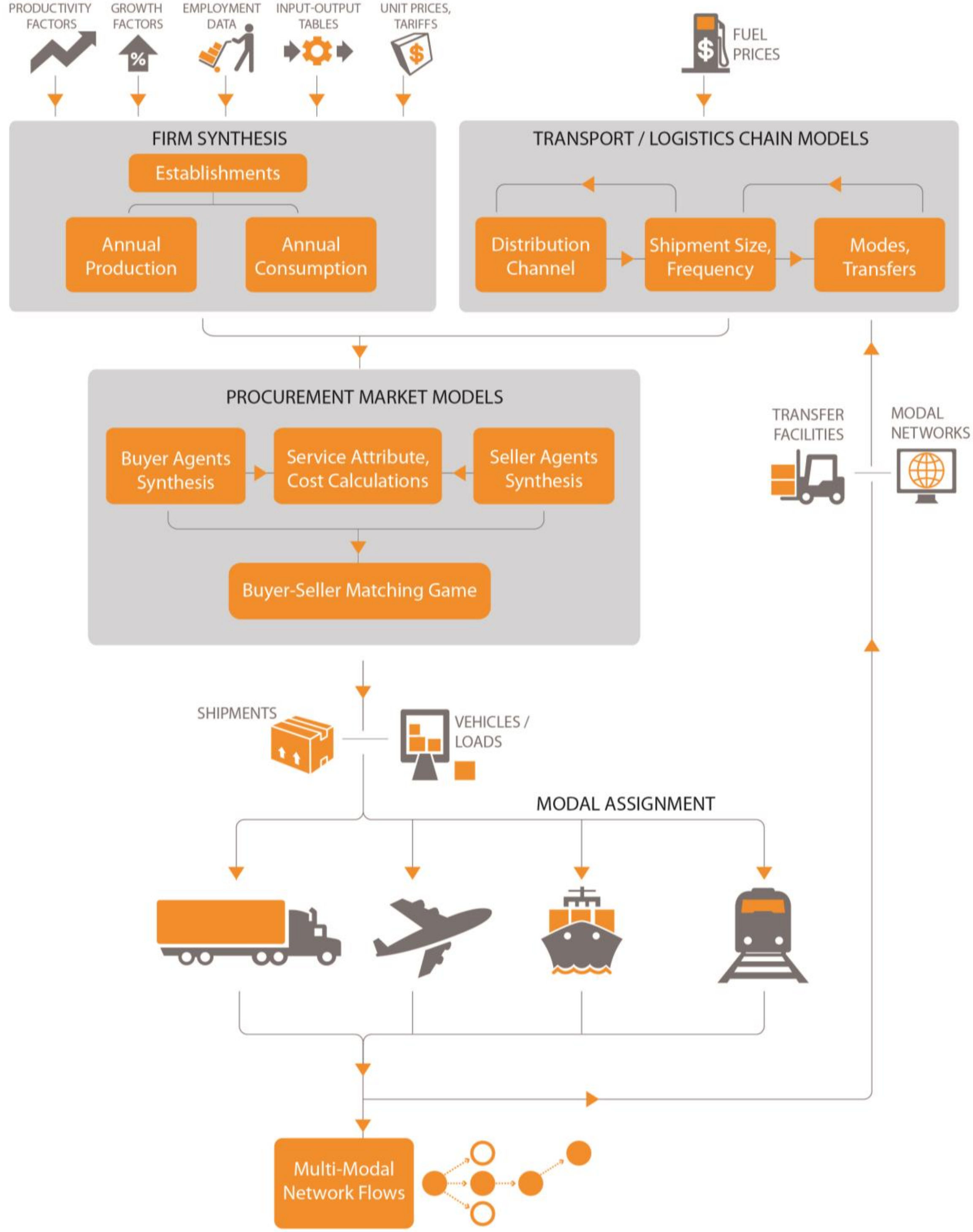

FIGURE 1: CMAP FREIGHT MODELING SYSTEM (rFREIGHT^{TM})

Although not yet fully implemented, the larger vision for the meso-scale freight modeling system is for the demand generated by shipments to be capacity constrained and thereby affect network level of service. Figure 1**Error! Reference source not found.** shows a feedback path in which assignment results update skims inputs to transport and logistics costs. Currently, only static service-flow attributes are used to inform the modal assignment.

For scenario analysis and forecasting future freight flows, rFreight™ allows the analyst to input assumptions regarding:

- Expected growth in domestic and foreign levels of consumption;
- Industry-specific productivity gains that affect IO relationships;
- Prospective changes to producer prices and tariffs on specific commodities; and
- Network-related parameters, such as assumptions on fuel costs and connectivity (port capacity expansions, intermodal terminal capacities, and similar infrastructure changes).

In the PMG kernel, it is also possible to change input parameters to reflect different assumptions regarding buyer tradeoffs between supplier cost and responsiveness and risk hedging. In addition, different sets of parameters and payoff weights may be specified to reflect assumed information available to agents and whether market prices are static or adjusted through game play.

The rest of this section describes each of these model components in more detail.

3.2 MACROECONOMIC ASSUMPTIONS

Several macroeconomic assumptions are implicit in the inputs to the model. These assumptions may be changed by modifying the inputs through processes that are described in Section 5.0. The baseline conditions for the model include the assumptions described below.

3.2.1 DOMESTIC PRODUCTION

The rates of commodity-specific domestic production (output) in the model are a function of the total number of domestic employees in the commodity-producing industries during the base year of the model (2010) and the commodity-specific total domestic production, as reflected in the BEA IO tables, for roughly the same period. This ratio of production to employment yields a rate of dollars of production per employee for each commodity-producing industry, and this rate applies uniformly to all domestic firms in the simulation. The model calculates these rates for each industry, internally, as described in Section3.5.2.

Analysis of future-year scenarios should account for changes in production rates by industry/commodity. Historically, changes in industrial productivity have come through automation and other technological advances. Trends in output per unit input for specific commodities or groups of commodities can be estimated by comparing BEA IO tables from different years. More current, albeit more aggregate, trends may also be derived from the U.S. Bureau of Labor Statistics' Multi-factor Productivity Indexes, which are published annually and show changes in output relative to labor, capital, and intermediate purchases.[1]

[1] http://www.bls.gov/mfp/

3.2.2 DOMESTIC CONSUMPTION

The rates of commodity-specific domestic consumption in the model are a function of output production quantities and the input requirements specified in the BEA IO ("Use") tables. The BEA tables provide a recipe for the fractional dollar value of various input commodities required to produce a dollar value of output. As described in Section 3.5.2, the model calculates these fractions internally by normalizing the consumption amounts shown in the IO tables by total industry production of the same commodity. For a given industry, total domestic production is equal to the sum of each row in the table. These recipes reflect the relationships observed/estimated by BEA for the period represented by the base year of the model.

Analysis of future-year scenarios should consider whether any changes in these IO relationships are likely to occur either because an input becomes relatively more or less costly, or due to substitution of input commodities. For example, the mix of energy sources (coal, petroleum, natural gas) used in manufacturing may change due to changes in availability and regulations. Less obvious changes might be substitution of new or different materials, such as composites used in construction that reduce the need for raw lumber.

3.2.3 LOCAL EMPLOYMENT

Due to the use of countywide totals in County Business Patterns, the allocation of simulated firms by industry type to the mesozones within the greater Chicago portion of the modeling system depends on the mix of employment by industry as observed for the base year of the model system. This allocation is based on the employment rankings file for each mesozone and industry type, which is described in more detail in Section 4.1.10. In essence, the assumption is that the same relative spatial distribution of employment that was present in the base year is valid for any scenario that uses this same file without changing it. Analysis of future-year scenarios should consider whether the Chicago region's distribution of employment by mesozone is likely to change from current patterns in a meaningful way.

3.2.4 IMPORTS AND EXPORTS

The volumes of imports and exports of commodities flowing to and from the United States and other countries are derived directly from the USA Census Trade Online (NAICS[2]-based data), provided by the U.S. Census Bureau's Foreign Trade Division.[3] These data are used to create representative, synthetic firms that compete in both buying and selling roles in U.S. commodity markets. The tables derived from this source, described in Sections 4.1.7 and 4.1.8, provide the total dollar value of commodities imported and exported to and from the U.S. for the baseline year of analysis.

For modeling purposes, rFreight™ assumes that the amount produced by each country for a particular commodity represents a fixed production capacity (supply), and ignores the demand-supply impacts of sales to non-U.S. countries. Similarly, rFreight™ assumes that the amount that the United States exports to each country represents a fixed demand for U.S. goods and ignores the demand-supply impacts of purchases from non-U.S. countries. These assumptions avoid the unnecessary complexity of modeling all possible world trade patterns. They effectively place upper bounds on import and export commodity flows that could be expanded by simply factoring up foreign production or consumption, which might be desirable for certain commodities

---

[2] North American Industry Classification System; This is the standard used by Federal statistical agencies in classifying business establishments for the purpose of collecting, analyzing, and publishing statistical data related to the U.S. business economy.

[3] https://usatrade.census.gov/

and countries. Analysis of future-year scenarios should consider whether the import and export relationships represented in the baseline model are likely to change and by how much.

3.2.5 PRODUCER PRICES

Producer prices used as attributes of seller agents in the model were derived from the Standard Classification of Transportable Goods (SCTG) data found in FAF3. For this version of the model, the NAICS 6-digit commodity codes were mapped to the SCTG 43 industry groups, providing a rough, aggregated approximation of producer prices, which were not available at the NAICS 6-digit level. These prices reflect U.S. producer prices for domestic goods and U.S. dollar values for imports and exports at the domestic port of entry or exit. The role of these prices in the model system is to set a nominal unit cost, which is added to transport and logistics costs and evaluated in the buyer-seller matching game, described in Section 3.6.

Producer prices are applied uniformly to all domestic manufacturers; therefore, they provide no discriminating information to the buyer-seller matching model when comparing the offerings of domestic producers. In future versions of the model, it may be possible to specify producer prices as a function of regional differences, firm size or other factors, and at more disaggregate level, consistent with the NAICS definitions.

Where these prices do make a difference in the current model is in comparing direct-manufacturer prices, domestic-wholesale prices, and import prices. Domestic firms that represent wholesaling of a commodity are assumed to charge a markup price, which is added to the SCTG unit cost. The assumption is that wholesalers provide offsetting storage, transport and logistics cost savings, which make them competitive on a local level. Nominally, a wholesale markup of 20% has been set and applied to all commodities offered for sale by wholesalers. In future versions of the model, it would be possible to vary wholesale markups by commodity type and, potentially, other variables such as firm size.

Foreign producers of each commodity are assumed to charge the SCTG cost, discounted to reflect typically cheaper foreign labor rates that attract U.S. firms to import commodities, offsetting longer shipping times and increased transport and logistics costs. Nominally, a 10% discount factor has been set and applied to all countries and all imported commodities. In future versions of the model, it would be possible to vary import prices factors by country, commodity type and, potentially, other variables.

Analysis of future-year scenarios should consider whether there is a need to change these pricing assumptions. In particular, the strength of the U.S. dollar relative to foreign currencies or foreign trade agreements that add or remove tariffs may be relevant considerations under certain scenarios and a reason to modify foreign discount factors.

In the current model, the prices offered by agents in the buyer-seller matching game do not affect the dollar-valued production or consumption rates, as described in Sections 3.2.1 and 3.2.2. The primary reason for this is that, out of necessity, they were derived from a different source than that used by BEA to develop the IO tables. In addition, while it may be theoretically possible to impose recursive consistency between the transaction prices offered/realized in the PMG market outcomes and the IO tables, the complexity of implementing such a market-clearing mechanism relegates it to future research and development. Accordingly, the baseline assumption is that the agents in the model are price-takers, meaning that the outcomes of their trades do not affect market production costs in the short run.

3.2.6 TRANSPORT AND LOGISTICS COSTS

Transport and logistic cost calculations are described in Section 3.4.4. The model assumes that the fuel prices (line-haul), storage costs and labor/handling cost components are the same as when the model parameters were estimated as a function of annual quantities and shipment weights as well as assumed discount factors and variations in performance attributes (lead times, damage, stock out). Analysis of future-year scenarios should consider whether these parameters are likely to change. In particular, it may be desirable to modify mode-specific line-haul charges to reflect changes in fuel prices.

3.3 SYNTHESIS OF FIRMS, ASSIGNMENT OF TYPES AND UTILITY PREFERENCE WEIGHTS

3.3.1 FIRM SYNTHESIS OVERVIEW

This section of the user's guide introduces the first major element of the model, depicted as "firm synthesis" in Section 3.3. The model uses many of the macroeconomic assumptions discussed above to control the synthesis of U.S. business establishments by industry classification and size. Business establishments are located spatially, and characterized with annual production and consumption requirements from existing commodity-flow relationships between producing and consuming sectors of the national economy, as represented in BEA IO accounts. The model system also synthesizes agents representing firms in countries and industries that currently trade with the U.S. The detailed step-by-step description of the firm synthesis model component is included below in Section 3.5.2.

3.3.2 PRODUCERS

Each firm that produces or sells (in the case of wholesalers) a transported commodity is identified as a producer. The total quantity (dollars) of that firm's primary production commodity, based on the 6-digit NAICS codes is calculated. The production quantities are generated using BEA estimates of total quantity produced per employee, multiplied by establishment size. The lists of producers are used to develop the sellers inputs to the PMGs, described in Section 3.6.4.

3.3.3 CONSUMERS

All business establishments consume commodities either as part of their production process to make the commodity that they produce, or otherwise in order to conduct their business (e.g., retailers do not produce goods, but do buy or consume the goods that they then sell from their establishments). The model bases the commodities that consumers use on the BEA estimates of consumption. The model considers the largest commodities by value that the establishment consumes, up to a threshold, in order to focus the simulation on the largest commodity transactions that each establishment makes. Section 3.6.4 describes the list of consumers used to develop the buyers' inputs to the PMG.

3.3.4 PREFERENCE WEIGHTS

The model assigns producers and consumers preference weights for shipping times and costs, based on the commodities that they trade. Based on the logistics literature, businesses are assumed to differ in the strengths of their preferences for costs savings versus responsive service. Commodities generally viewed as "functional" are considered to have a low value-to-weight ratio, and thus cost savings is proportionally more important than responsiveness. Commodities that are generally viewed as "innovative" are generally considered to have a high value-to-weight ratio, and thus responsiveness is proportionally more important than thus cost savings. For similar reason, it is assumed that buyers attempt to minimize risk by spreading their sourcing contracts to multiple suppliers, more so for innovative commodities.

SCTG commodity groups are categorized into four commodity types (functional/innovative/semi-functional/semi-innovative) based on commodity demand, cost, and time-sensitivity characteristics. These are the main supply chain related attributes of a commodity that affect the logistics decisions and defines product type. The cost characteristics are based on FAF commodity-flow data dollar/ton values for all modes. Demand uncertainty and time-sensitivity characteristics are based on judgment and the nature of the product, assuming four score levels of uncertainty and sensitivity (very low/low/medium/high). The total score level for each commodity is then calculated using the assigned score levels for each commodity characteristic score. Finally, the total score levels are used to categorize commodity groups into the four types mentioned above. For preference weights, it is assumed that for functional products more weight is on cost saving (generally needs an efficient supply chain) and for innovative products more weight is on time saving (generally needs a responsive supply chain). For semi-functional and semi-innovative products, the cost and time preference weights are more equally important, but marginally shifted toward cost-saving and time-saving respectively.

The model uses the data_firm_pref_weights.csv input files to associate establishments with particular values for these weights.

3.4 CHOICE OF SHIPMENT SIZES AND MODES—CALCULATION OF COSTS AND SHIPMENT TIMES

3.4.1 TRANPORT/LOGISTICS CHAIN OVERVIEW

This section of the user's guide introduces the second major element of the model, depicted as "transport/logistics chain models" in Section 3.4. This part of the model develops shipping times and costs for each of the possible combinations of buyers and sellers to be evaluated in the PMGs so that transportation costs can play a role in the trading decisions made by buyers and sellers. The outputs from this element of the model are used to develop the costs inputs to the PMGs, described in Section 3.6.4.

3.4.2 DISTRIBUTION CHANNEL MODEL

The logistics chain models begin with a model that selects the distribution channel for the shipment, a key business decision made by shippers. A distribution channel refers to the supply chain a particular shipment follows from the supplier to the consumer/buyer and is critical to business freight related operations. The supplier firms may use their own transportation resources or send shipments to the buyer using third-party logistics firms. The distribution channel might affect the cost, shipment size and frequency of shipments between a buyer-supplier firm pair.

The Freight Activity Microsimulation Estimator (FAME) establishment survey collected by University of Illinois at Chicago was used to represent data on the supply chain, which contained data on whether the goods went through a consolidation center, a distribution center, and/or a warehouse. The concept of distribution channel was simplified to obtain a reasonable sample for model estimation. Four alternatives for distribution channel were identified – direct, one stop type, two stop types, and three stop types where stop type is a warehouse, distribution center, or consolidation center. The variables that affect the choice of distribution channel in a significant manner are firm size and the industry type of the firms involved.

Models were estimated for manufactured goods and for food products. The distribution channel choice was simulated for shipments between all the buyer-supplier pairs based on the type of commodity. For commodities other than food or manufactured good, the manufactured goods model was applied.

3.4.3 SHIPMENT SIZE MODEL

In the shipment size model, the likely shipment size (weight) of shipments are determined. Shipment size affects the mode used to transport the shipment. A multinomial logit model was estimated for choice of shipment size. The Texas commercial vehicle survey dataset was used for estimating the discrete choice model due to its relatively high sample size. This dataset is not ideal for the shipment size model because the shipments represented in the dataset are likely to be within an urban area; however, it was the most appropriate dataset, considering the smaller sample sizes in other available datasets. Three alternatives form the choice set – less than or equal to 999 lbs., 1000-9999 lbs., and greater than 9999 lbs.

Separate models were estimated for food and manufactured products. For food products, the shipment size between 1000 and 9999 lbs. is preferred most, all else being equal. Longer trip lengths are associated with shipments greater than 10,000 lbs. For manufactured products, the explanatory variables are similar to those in the food products model. Shipments less than or equal to 999 lbs. seem to be preferable, everything else being equal. Here, longer trip lengths are associated with shipment sizes less than or equal to 999 lbs.

Simulation of shipment size results in the assignment of a shipment to one of the three broad shipment categories. To obtain a more defined shipment size each size category was split into bins. Probability thresholds for a shipment being in one of ten bins were calculated from Texas survey data and Commodity Flow Survey data. Each shipment was then assigned to one of the finer shipment size categories, using Monte Carlo simulation.

3.4.4 MODE CHOICE MODEL

The freight mode choice model assigns a mode for shipments transported between each possible buyer-supplier pair. There are four primary modes (road, rail, air, and water) modeled. The model code and model network for this element were developed by Cambridge Systematics (2011) for the CMAP Meso-scale Freight Model and was adapted by RSG for the FHWA freight forecasting framework (RSG, 2012). The work by Cambridge Systematics adopted methods from de Jong and Ben-Akiva (2007) to predict the path and mode of long-haul movements of freight into, out of, within, and through the CMAP modeling area. The path includes identifying the location of intermodal transfer facilities, distribution centers, or warehouses where shipments are consolidated or de-consolidated. Detailed networks of road and rail for the U.S. were used, but detailed networks of air and water were estimated using simple functions of distance and the value of goods being transported. The model for mode choice and intermodal transfers is based on the formulation developed by de Jong and Ben-Akiva (2007):

$$G_{mnql} = \beta_{0ql} + \beta_1 * \left(\frac{Q}{q}\right) + T_{mnql} + \beta_2 * j * v * Q + \beta_3 * t_{mnl} * v * \frac{Q}{365} + (\beta_4 + \beta_5 * v)\left(\frac{q}{2}\right) +$$

$$\beta_5 * v * a * \sqrt{LT * \sigma_Q^2 + Q^2 * \sigma_{LT}^2}$$

Descriptions of variables and parameter notations may be found below in Section 3.5.3. Discount rates are based on the type of commodity being transported. Bulk natural resources have a low-discount rate. Animals and intermediate processed goods have a medium-discount rate. A high-discount rate is applied for finished goods.

Mode-path skims (times and costs) were developed for 54 alternatives that include direct modes (such as truck – Full truck load, truck – less than truck load, rail, etc.); indirect mode (such as rail-truck, water-truck etc.); and also intermodal facilities (such as airports, truck terminals, rail terminals, and ports). The skims used are described in Section 4.1.13.

Based on the distribution channel the choice set of mode-path evaluated is restricted. For firms pairs using a direct distribution channel, only direct mode-paths are evaluated – Full Truck Load (FTL) Direct, Less than Truck Load (LTL) Direct, Carload Direct (rail), and Intermodal Marketing Extension (IMX) Direct (rail). The remaining 50 mode-paths are evaluated for firms using indirect distribution channels. Once the generalized cost is evaluated for all the alternatives for each buyer-supplier pair, the least-cost alternative is chosen as the mode-path. The shipment time and cost for this mode-path is then used in the costs file input to the PMGs.

3.5 MODEL PROCESSING STEPS

3.5.1 MODEL STRUCTURE

This section of the user's guide explains what occurs at each step within the model system (i.e., rFreight™ the R code that incorporates the PMG application). The discussion covers the data manipulation and calculations that are occurring at each stage, as well as cross-referencing to the inputs that are used and the outputs that are produced.

The model is comprised of a set of folders, inputs files, output files (once the model is run), R scripts, and the PMG application. Table 1 shows the files and folders in the top level of the model folder and explains the purpose of each one.

TABLE 1: THE MODEL'S TOP LEVEL FOLDERS AND FILES

FILES	TYPE	PURPOSE
check	Folder	Check folder for use during compiling the rFreight™ package
chicago-agent.Rproj	File	R project file for use with R Studio
CMAP_Agent_Run.R	File	R script file used to define which scenario to run and execute model
PMG	Folder	Folder containing the PMG application and source code
rFreight	Folder	Folder containing the source code for the rFreight™ package
rFreight_0.1.zip	File	Zip file of the compiled rFreight™ package
scenarios	Folder	Folder containing model scenarios (inputs, parameters, outputs)
scripts	Folder	Folder containing the model's R scripts

Table 2 lists the R scripts in the scripts folder and explains the purpose of each. The structure inside the scenarios folder, which contains the inputs, parameters, and outputs from each scenario, is discussed in Section 4.0.

TABLE 2: FILES IN THE SCRIPTS DIRECTORY

FILES	TYPE	PURPOSE
0_Create_rFreight.R	Package Creation	Documents, checks, and builds the rFreight™ package
0_Main.R	Model Control	Overall control, called from CMAP_Agent_Run.R, calls other scripts in sequence
0_File_Locations.R	Model Control	File path definitions
01_Firm_Synthesis.R	Model Component	Produces synthesized firm list
02_Procurement_Markets.R	Model Component	Creates workspace containing functions for running each market separately
03_PMG_Controller.R	Model Component	Script to loop over markets and start separate processes
03a_Run_PMG.R	Model Component	Script run in separate R script instances to set up and run PMGs
03b_Monitor_PMG.R	Model Component	Script run in separate R script instance to monitor and report on progress
04_PMG_Outputs.R	Model Component	Script to loop over and reassemble the outputs
05_Daily_Sample.R	Model Component	Script take a sample from the overall table for local Chicago/daily simulation
06_Warehouse_Allocation.R	Model Component	Script to assign warehouses in Chicago region
07_Vehicle_Choice_Tour_Pattern.R	Model Component	Script to apply the vehicle and tour choice model
08_Stop_Sequence.R	Model Component	Script to apply the number of tours model, the clustering model, and the stop sequence model
09_Stop_Duration.R	Model Component	Script to apply the stop duration model
10_Time_of_Day.R	Model Component	Script to apply the time of day model and also to split long tours
11_Prepare_Trip_Table.R	Model Component	Script to aggregate trips into trip tables

The overall model flow is controlled by the 0_Main.R script in the scripts folder. To initiate a scenario run (following completing the development of inputs and setting the parameter values to use), the model user edits the CMAP_Agent_Run.R script to identify where the model is located on the model user's computer and which scenario should be run, and then runs the three lines of code in the script. That script then passes control to 0_Main.R, which runs the remainder of the model. 0_Main.R has two parts, which deal with model set up and model component execution, respectively.

In the model set up portion of the script, 0_Main.R first installs and loads the rFreight™ R package. This is an R package written specifically for this model, which contains many of the R functions used by the model to control model flow, track progress, and execute core modeling functions such as simulating logit models. The documentation for the rFreight™ package is included as an Appendix to this user's guide. The 0_Main.R script then calls the startModel function to create objects in memory that hold model and scenario structure information, to load scenario variables and other required R packages, and to create additional objects to hold information about each model component. The remaining code in this section of 0_Main.R reads in and organizes the file locations for each model component's inputs and outputs.

In the model component execution portion of the script, 0_Main.R first calls the progressManager function to start aspects of model management such as log files, recording run times, and progress bars, and then sequentially calls each of the model components, which are contained in separate scripts, numbered from 1 to 11. Once the model has executed all 11 model components, the progressManager function is called again to close down logging and run-time recording and exit the model. The 11 model components are briefly introduced in the table above and discussed in more detail in the rest of this section..

The third component of the model, which controls, monitors and runs each individual PMG, is split into three scripts. The 03_PMG_Controller.R script executes in the main model thread, run from R Studio. This script uses the system function in R to start the 03b_Monitor_PMG.R script in an instance of Rscript.exe in a separate thread, and then makes calls via the system function to start multiple instances of Rscript.exe running the 03a_Run_PMG.R script. This process, discussed in more detail below, is designed to make use of the multiple processors available on the model user's computer to reduce model run time.

As noted above, the model uses several objects held in memory to provide organization of the model structure, and hold lists of variables, inputs, and outputs. These objects use R's list data type, which provides a flexible data structure that can hold items of different types in each position in the list (e.g., individual values, vectors, tables, or other lists). At the end of the model run, the complete set of these lists are saved in the outputs folder for the scenario in the modellists.R data file.

The model object holds information about the overall model structure and settings for the particular scenario run during which it was created. Table 3 describes the contents of the model object, and Table 4 describes the list structure used to organize inputs, outputs, and summaries.

TABLE 3: STRUCTURE OF THE MODEL OBJECT

LIST ITEMS	DESCRIPTION
basedir	Model directory path
scendir	Scenario directory path
scenvars	Scenario variables (read in from scenario_variables.R)
inputdir	Input directory path
outputdir	Output directory path
scriptsdir	Scripts directory path
packages	R packages loaded to support running the model
steps	Short names of model steps used for component lists
steptitles	Long names of model steps used for progress bars and logging text
stepscripts	File paths for model component scripts
sessionInfo	Output from sessionInfo() function providing version information about R and attached or loaded packages
workspace	File path for saving the model's workspace
logs	File paths for saving model logs

TABLE 4: GENERIC STRUCTURE OF MODEL COMPONENT OBJECTS

LIST ITEMS	DESCRIPTION
step	Short model component name
steptitle	Long model component title
inputs	Input object names and file names
inputtables	List to hold input tables
summary	Summary file names
summaryfunc	Summary function to produce summaries
outputs	Tabular outputs
validation	Validation inputs
workspace	File path for saving the model's workspace

3.5.2 COMPONENT 1: FIRM SYNTHESIS

Description

This is the first step in the model and involves synthesis of individual firms or businesses using aggregate control data and characterization of the firms as either (or both) buyers and sellers of various commodities.

Inputs and Parameters

Table 5 and Table 6 show the input tables and parameters, respectively, used in the firm synthesis step of the model. The input tables are read into memory and held in objects named as shown in the "object name" column of Table 5. Details of the layout of the input tables are described in Section 4.1.

TABLE 5: INPUTS TO THE FIRM SYNTHESIS STEP

FILE NAME	OBJECT NAME	DESCRIPTION
corresp_naics6_n6io_sctg.csv	c_n6_n6io_sctg	Correspondence between NAICS categories and SCTG commodities categories
data_2010io.csv	io	Detailed 2010 Use table after redefinitions data developed using the 2007 benchmark IO account
data_emp_cbp.csv	cbp	Number of employees and establishments by six digits NAICS industry, FAF zone, and county
data_firm_pref_weights.csv	prefweights	Cost and time preference weights by SCTG commodity categories
data_foreign_cons.csv	for_cons	Total exports value, valued on a free alongside ship (FAS) basis, by country and 6-digit NAICS code
data_foreign_prod.csv	for_prod	CIF (cost, insurance, and freight) value of US imports by country, and 6-digit NAICS code
data_mesozone_emprankings.csv	mzemp	Employment ranking by industry by county
data_unitcost.csv	unitcost	Unit cost (dollar/ton) by two digits SCTG commodity codes

TABLE 6: PARAMETERS USED IN THE FIRM SYNTHESIS STEP

PARAMETER	BASE VALUE	DESCRIPTION
Provalthreshold	8e-01	Threshold for percentage of purchase value for each commodity group met by producers
combinationthreshold	7e+06	Maximum number of combinations of producers and consumers to enter into a PMG
Consprodratiolimit	1e+06	Limit on ratio of consumers to producers to enter into the PMG
foreignprodcostfactor	9e-01	Producer cost factor for foreign producers (applied to unit costs)
wholesalecostfactor	1.2	markup factor for wholesalers (applied to unit costs)

Model Steps

The firm synthesis component of the model applies the following steps:

1. ***Creating Firm Records*** -- The firm synthesis model begins by enumerating business establishments using pre-processed aggregate control data from the Census County Business Pattern (CBP) data, creating a list of firms.
   a. A correspondence between NAICS6 (6-digit NAICS), the NAICS used in I-O tables (NAICSIO), and SCTG is used to assign NAICSIO and SCTG commodity produced designations for each firm.
   b. The dataset is enumerated (one record per business establishment), where each firm is identified by NAICS and NAICSIO, CBPZone, FAFZone, size category, employment (category mid-point), and a unique sequential business ID starting at 1.
   c. Some industries produce more than one commodity, such as wholesalers. To account for this, the production of one commodity for each such business establishment is simulated, based on the probabilities of the multiple commodities that it could produce.
2. ***Placing Businesses within Mesozones*** -- Until this point, the geographic identifier for all business establishments is the CBPZone, based on county boundaries. The next step is to locate business establishment within the higher-resolution geographic unit, mesozones.
   a. From the database of all business establishments, select businesses in the CMAP modeling area. There are a few counties which correspond to only one mesozone. The firms in these counties are directly assigned a mesozone.
   b. The other counties in CMAP modeling area are composed of more than one mesozone. Firms in these counties are assigned to mesozones based on employment ranking by industry. A pre-processed dataset contains the percentile ranking of each of 21 NAICS categories by mesozone, based on employment numbers in each of those industries. Higher employment implies a higher percentile rank; this rankings dataset is merged with the business establishment database by county.
   c. For each of the 21 NAICS categories considered, candidate mesozones are identified based on business establishment size and the ranking of a particular NAICS in a mesozone. The probability of a mesozone getting assigned to a particular business establishment increases with the rank of the business establishment's NAICS in the mesozone and the number of employees in the business establishment. For example, if a business establishment belongs to

the manufacturing industry and has a size greater than 5,000 then all mesozones which have manufacturing ranked 9th or 10th are candidates for the particular firm.

   d. Once candidate mesozones are assigned to each business establishment, one of the candidates is randomly selected as the business establishment's mesozone.
   e. Business establishments not in the CMAP modeling area are assigned a mesozone number based on their CBPZone number and combined with the CMAP modeling area business establishment dataset to create a full business establishment database with mesozones attached.

3. ***Warehouse Locations*** are saved for use in the warehouse allocation model component. This list is composed of firms in the CMAP modeling area with the NAICS codes 481, 482, 483, and 493.
4. ***Producers/Suppliers Database*** -- all firms that produce a commodity with an SCTG code (i.e., one that requires transportation).
   a. The model uses the IO data to create a table of production values and employment by NAICSIO code to calculate production value per employee, as the sum of commodity production by NAICSIO from the IO data divided by the sum of employment from the producer database.
   b. The production value per business establishment is its employment multiplied by the production value per employee.
   c. Until this point, the model has dealt with domestic producers. The import data is used to develop a list of foreign producers of each imported commodity for each country from which imports of that commodity are observed.
   d. The unit cost input is used to convert foreign production values into production capacities in pounds.
   e. The *foreignprodcostfactor* variable is used to scale the employment and production capacities of foreign firms to account for different employment intensity/productivity and production prices.
   f. Similarly, the unit cost input is used to convert domestic production values into production capacities in pounds.
   g. A single producer's database including both domestic and foreign producers is created, with fields that are not needed removed and naming consistent with the naming requirements for the PMG application.
5. ***Consumers/Users Database*** -- all the firms in the firm database merged with processed IO data based on the NAICSIO of the firm as a consumer.
   a. The processed IO table identifies the commodities (in terms of production value) consumed by each consumer industry NAICSIO, normalized by total consumption of all commodity inputs for that industry. The processed IO data removes the smaller value commodities consumed by a given industry so that the amount of consumption recognized just exceed the *provalthreshold*. In other words, with a *provalthreshold* set at 80%, the consumer database will list commodity requirements for each industry up to just over 80% of total consumption, and will do this based upon the smallest possible number of separate inputs. This is done to reduce the complexity of the simulation, where the many small commodity inputs that make up the remaining percentage of requirements above the *provalthreshold* are ignored.

   b. Similar to what is done for producers, the SCTG commodity for suppliers who could produce more than one SCTG commodities is simulated using probability thresholds. This is done for all the suppliers being considered.
   c. The consumption value of each commodity for each business establishment is calculated using the value per employee calculated earlier.
   d. This is converted to purchase amounts in pounds using the unit costs.
   e. A set of foreign consumers are developed, for each combination of country and export commodity identified in the trade data. They are characterized with purchase requirements in a similar way to the domestic consumers, and added to the list of domestic consumers to create a complete consumers database.
   f. The extra variables required by the PMGs, e.g., preference weights, are added to the consumers database, unneeded fields are removed, and naming is consistent with the naming requirements for the PMG application.

6. ***Commodity Market Sampling and Grouping*** -- With the producers and consumers tables complete, appropriate sample sizes for the groups within each commodity market are calculated in such that the number of combinations of producers and consumers placed in each group is less than the *combinationthreshold*, while the *consprodratiolimit* is respected.
   a. In commodity markets where there are many consumers for every producers, the same producer is entered into multiple groups with a portions of its production allocated into each group.
   b. At the end of firm synthesis, the model writes out an R workspace for each commodity market into the outputs folder, containing tables of the producers and consumers of that particular commodity.

Outputs

Table 7, below, shows the outputs that are produced by the firm synthesis step of the model. Additional details such as the layout of output tables are presented in Section 4.3.

TABLE 7: MODEL OUTPUTS FROM THE FIRM SYNTHESIS STEP

FILENAME	TYPE OF OUTPUT	DESCRIPTION
XXXXXX.Rdata	R Binary	A file for each NAICS market (where XXXXXX is the 6-digit NAICS code) containing the tables of producers and consumers for each market, which is output from firm synthesis. The inputs to the individual PMGs are built from these files. (The pairs table of final trades for all of the PMGs covering this market is added to this Rdata file after the PMGs are run)
Firm_syn.txt	Summary	A summary text file showing tabulations from the firm synthesis model
Naics_set.Rdata	R Binary	A summary of the numbers of producers, consumers, matches, and numbers of PMG groups for each NAICS market
Warehouses.Rdata	R Binary	A list of warehouse locations summarized from the firm synthesis for use in the warehouse allocation model

3.5.3 COMPONENT 2: PROCUREMENT MARKETS

Description

This model component creates an R workspace file containing functions for running each market separately. These functions are focused on dividing the lists of producers and consumers into appropriate groups for simulation in the PMGs and creating the costs inputs to the PMG for the combinations of producers and consumers in each group.

Inputs and Parameters

Table 8 and Table 9 show the input tables and parameters that are used in the procurement market step of the model. The input tables are read into memory at the start of this step of the model and held in objects named as shown in the object name column of the table. Details of the layout of the input tables are described in Chapter 4.

TABLE 8: INPUTS TO THE PROCUREMENTS MARKET STEP

FILENAME	OBJECT NAME	DESCRIPTION
corresp_sctg_category.csv	sctg	Correspondence between SCTG categories and commodity
data_mesozone_gcd.csv	mesozone_gcd	Great circle distance (GCD) between mesozones
data_modepath_skims.csv	skims	Skim costs and times for set of multimodal paths
model_distchannel_calibration.csv	distchan_cal	Distribution channel type (number of stops, 0, 1 or 2+ stops) model calibration data: shares by commodity groups
model_distchannel_food.csv	distchan_food	Distribution channel model variables and coefficients by distributing channel type for food products
model_distchannel_mfg.csv	distchan_mfg	Distribution channel model variables and coefficients by distributing channel type for manufactured products
model_shipsize_calibration.csv	ShipSize_cal	Shipment size model calibration data: shares by shipment weight groups and commodity groups for value and tons
model_shipsize_food.csv	ShipSize_food	Shipment size model variables and coefficients by shipment size groups for food products
model_shipsize_mfg.csv	ShipSize_mfg	Shipment size model variables and coefficients by shipment size groups for manufactured products

TABLE 9: PARAMETERS USED IN THE PROCUREMENT MARKET STEP

PARAMETER	BASE VALUE	DESCRIPTION
B1	100	Constant unit per order
B4	2000	Storage costs per unit per year
j	0.01	Fraction of shipment that is lost or damaged
LT_OrderTime	10	Expected lead time (time between ordering and replenishment)
sdLT	1	Standard deviation in lead time
BulkHandFee	1	Handling charge for bulk goods ($ per ton)
WDCHandFee	15	Warehouse/DC handling charge ($ per ton)
IMXHandFee	15	Intermodal lift charge ($ per ton)
TloadHandFee	10	Transload charge ($ per ton; at international ports only)
AirHandFee	20	Air cargo handling charge ($ per ton)
WaterRate	0.005	Line-haul charge, water ($ per ton-mile)
CarloadRate	0.03	Line-haul charge, carload ($ per ton-mile)
IMXRate	0.04	Line-haul charge, intermodal ($ per ton-mile)
AirRate	3.75	Line-haul charge, air ($ per ton-mile)

PARAMETER	BASE VALUE	DESCRIPTION
LTL53rate	0.08	Line-haul charge, 53 feet LTL ($ per ton-mile)
FTL53rate	0.08	Line-haul charge, 53 feet FTL ($ per ton-mile)
LTL40rate	0.1	Line-haul charge, 40 feet LTL ($ per ton-mile)
FTL40rate	0.1	Line-haul charge, 40 feet FTL ($ per ton-mile)
ExpressSurcharge	1.5	Surcharge for direct/express transport (factor)
BulkTime	72	Handling time at bulk handling facilities (hours)
WDCTime	12	Handling time at warehouse/DCs (hours)
IMXTime	24	Handling time at intermodal yards (hours)
TloadTime	12	Handling time at transload facilities (hours)
AirTime	12	Handling time at air terminals (hours)
LowDiscRate	0.01	Low-discount rate
MedDiscRate	0.05	Medium-discount rate
HighDiscRate	0.25	High-discount rate
CAP1FTL	60000	Truckload capacity (pounds)
CAP1Carload	170000	Carload capacity (pounds)
CAP1Airplane	50000	Air cargo hold capacity (pounds)
LowMultiplier	0.5	Safety stock constant for Low category commodities
MediumMultiplier	1	Safety stock constant for Medium category commodities
HighMultiplier	2.33	Safety stock constant for High category commodities
LowVariability	0.03	Standard deviation in annual flow for Low category commodities
MediumVariability	0.06	Standard deviation in annual flow for Medium category commodities
HighVariability	0.09	Standard deviation in annual flow for High category commodities

Model Steps

The procurement market component of the model applies the following steps:

1. The *create_pmg_sample_groups* created in this model component is used to divide producers and consumers into groups subject to the sample sizes calculated at the end of the firm synthesis component. The function includes check and resampling routines to ensure that the total production capacity in each group is greater than the purchase requirement of the consumers.
2. The *create_pmg_inputs* function develops the three input files for the PMG from the list of producers and consumers assigned to each group. In addition to writing out the producer and consumer inputs,

the function applies three models to develop the cost inputs to the PMG: a distribution channel model, a shipment size model, and a mode-path choice model.

3. This distribution channel model simulates the channel that would be used by each possible combination of producers and consumers (pairs). Four alternatives for distribution channel are simulated: direct, one stop type, two stop types, and three stop types where stop types are warehouse, distribution center, or consolidation center. Separate models were estimated for food and manufacturing products. For other commodities, the manufactured products model is applied.
4. Extra variables required by the two distribution channel model specifications are created from the characteristics of the producers and consumers and the shipment.
5. A logit simulation function simulates the distribution channel for each pair involving food products and pairs involving all other commodities.
6. Simulation of the shipment size is also based on a logit model. There are three alternatives: less than 999 lbs., 999-9999 lbs., and greater than 9999 lbs.
7. Separate shipment size models were estimated for food and manufacturing products. As in the case of distribution channel model, for commodities other than food and manufactured products, the manufactured products model is applied.
8. Extra variables required by the shipment size model specifications are created.
9. The logit simulation function is applied to determine the range of the shipment size used by each pair of producers and consumers.
10. Once the shipment size range is known, another function is used to simulate the exact shipment weight based on probability thresholds for each shipment size category.
11. The mode-path choice model determines the mode and path of the shipments between each firm pair. There are 54 mode-path combinations, of which four are direct (two truck and two rail mode-paths). The skim dataset has times and costs associated with all the 54 mode-path choices between all mesozone pairs. For mesozone pairs not covered by the skims (i.e., without an origin or destination in the CMAP region), the skims are simulated based on zone-to-zone distances and modal speeds.
12. A commodity category and SCTG code correspondence is merged into the pairs datasets.
13. Annual logistic costs for each pair is calculated stochastically based on skims, mode-path capacities, storage costs, damage/loss costs, etc., for all the available mode-path alternatives. Four mode-paths are evaluated for direct distribution firm pairs, and 50 mode-paths are evaluated for the indirect distribution firm pairs.
14. The least-cost mode-path is selected for all the firm pairs. For a few pairs of firms with a direct distribution channel, direct skims are unavailable. In those cases, the indirect mode-path skims are used and an indirect mode-path is selected. The distribution channel is also changed from direct to an indirect one based on the proportions of occurrence. The opposite is done if indirect skims are unavailable for a firm pair with an indirect distribution channel.
15. The *create_pmg_inputs* function saves the full pairs table for use later in the model in an R workspace and writes out the cost.csv file for use an input to the PMGs.
16. The functions created in this model component are saved into the PMG_Workspace.R data file, which is loaded in by each of the separate Rscript.exe processes started in the next model component.

Outputs

Table 10, below, shows the outputs that are produced by the procurement markets step of the model. Additional details such as the layout of output tables are presented in Section 4.3.

TABLE 10: MODEL OUTPUTS FROM THE PROCUREMENT MARKETS STEP

FILENAME	TYPE OF OUTPUT	DESCRIPTION
PMG_Workspace.Rdata	R Binary	A workspace containing function, variables and other data from the early portion of the R model that is used in each of the separate R script processes to support development of the inputs to the PMGs

3.5.4 COMPONENT 3: PMG CONTROL, EXECUTION, AND MONITORING

Description

In this component, the main R thread that is running the model starts a series of separate instances of Rscript.exe, with one monitoring and reporting on overall progress and the others running each individual commodity market -- creating the separate inputs to the PMGs, calling the PMG application, and processing the results from the PMGs.

Inputs and Parameters

Table 11 shows the parameter that is used in the PMG control step of the model. The parameters used in the PMGs themselves (i.e., those that comprise the PMG .ini file) are discussed earlier in this chapter. There are no new inputs used in this step of the model; all data used are products of the earlier model components.

TABLE 11: PARAMETERS USED IN THE PMG CONTROL STEP

PARAMETER	BASE VALUE	DESCRIPTION
maxrscriptinstances	16	Maximum number of separate R scripts that can be run at one time; set subject to limit of computer's available RAM
pmgmonitoring	TRUE	Should monitoring be activated during running the PMGs to email updates to the model users?
pmgmonfrom	"colin.smith@rsginc.com"	PMG monitoring email (from email address)
pmgmonto	"colin.smith@rsginc.com", "john.gliebe@rsginc.com"	PMG monitoring email (to email address)
pmgmonsmtp	"WRJHUBVPW01.i-rsg.com"	PMG monitoring email transmission server information
pmgmoninterval	3600	PMG monitoring interval (seconds)
pmglogging	TRUE	Should pmg logging be on?

Model Steps

The PMG control component of the model applies the following steps:

1. The model calls the *writePMGini* function to create the ini file to be used for the PMGs, based on the scenario variables that were read in earlier in the model run
2. If *pmgmonitoring* is set to TRUE, the model starts a separate Rscript.exe process and passes the 03b_Monitor_PMG.R script to that process. This script monitors the progress of the run. Periodically, it looks into the outputs folder at the log files and PMG output files and creates a summary of the in progress and completed PMGs. This summary can be emailed to the model user as an attachment for the purpose of monitoring progress without needing to look at the computer where the model is running. Several parameters used by this script require tailoring to the analyst's computer and email address in order for emails to be sent and received: *pmgmonitoring, pmgmonfrom, pmgmonto, pmgmonsmtp, pmgmoninterval*.
3. Depending on the value of the *maxrscriptinstances* and *pmgmonitoring* parameters, the model starts several separate Rscript.exe processes (*maxrscriptinstances* -1, if *pmgmonitoring* is set to TRUE and one process is used for the 03b_Monitor_PMG.R script). The user should set this parameter based on the amount of RAM installed on their computer – each process will use up to around 4GB when the base value of the *combinationthreshold* parameter is used (that parameter controls how the size of each individual PMG.)
4. The model passes the 03a_Run_PMG.R script to each of the Rscript.exe processes. The 03a_Run_PMG.R script executes the PMG for a single commodity.
5. The PMG workspace prepared in the second model component is loaded along with the producers and consumers tables for the commodity being run.
6. The processes described above in the second model component are now executed for the commodity being run: the producers and consumers are divided into sample groups, and for each group, the PMG inputs files are created.
7. The PMG application is called for each group.
8. The .csv inputs files are deleted for each group upon completion of the PMG application.
9. The output from the PMG application is read in and the trades from the final iteration of the PMG are joined back onto information about the buyers and sellers and combined into a single table for all of the sample groups for this commodity.
10. Once the results for the commodity are saved and the 03a_Run_PMG.R script completes, the Rscript.exe process closes and the model starts a new Rscript.exe progress to run the next commodity in line to be run.
11. Once the model has started all of the commodity groups in separate Rscript.exe processes, this component of the model is completed, and the next component (PMG outputs) starts.

Outputs

Table 12, below, shows the outputs that are produced by the PMG control, execution and monitoring step of the model. Additional details such as the layout of output tables are presented in Section 4.3.

TABLE 12: MODEL OUTPUTS FROM THE PMG CONTROL, EXECUTION, AND MONITORING STEP

FILENAME	TYPE OF OUTPUT	DESCRIPTION
XXXXXX.Rdata	R Binary	A file for each NAICS market (where XXXXXX is the 6-digit NAICS code) containing the tables of producers and consumers for each market, which is output from firm synthesis. The inputs to the individual PMGs are built from these files. The pairs table of final trades for all of the PMGs covering this market is added to this Rdata file after the PMGs are run
XXXXXX.txt	Log file	Log file showing start and end times for each PMG run for this NAICS market (where XXXXXX is the 6-digit NAICS code)
XXXXXX_PMG_Log.txt	Log file	More detailed log file showing the print messages produced during the creation of PMG inputs, calls to run the PMG model, and processing of outputs for this NAICS market (where XXXXXX is the 6-digit NAICS code)
XXXXXX_gY.csv	PMG output	Output csv file from a particular PMG run for a NAICS market (where XXXXXX is the 6-digit NAICS code) and group, where Y is the group number

3.5.5 COMPONENT 4: PMG OUTPUTS

Description

The PMG outputs component of the model reassembles the individual commodity market results from the PMGs into a single large tabular output that can be used in subsequent portions of the model.

Inputs and Parameters

There are no new inputs or parameters used in the PMG outputs component of the model.

Model Steps

The PMG outputs component of the model applies the following steps:

1. The model loops over each of the commodity markets that are being run in this scenario and looks for whether the results are complete yet.

2. As the results are complete and available, the model loads the commodity workspace file from the results folder and adds the final pairs table containing just the trades from the final iteration of the PMGs to a list of results from all of the commodity markets.
3. Once the results are available from all of the commodity markets that are being run in this scenario, the model collapses the list of results into a single pairs table now containing all of the trades from the final iteration of the PMGs. This pairs table is saved (as the final result of the first portion of the model that simulates trade relationships, commodity flows, and shipments), and is passed to the second portion of the model that simulates regional truck tours to pick up and deliver shipments in the CMAP region.

Outputs

Table 13 shows the outputs that are produced by the PMG output step of the model. Additional details such as the layout of output tables are presented in Section 4.3.

TABLE 13: MODEL OUTPUTS FROM THE PMG OUTPUT STEP

FILENAME	TYPE OF OUTPUT	DESCRIPTION
Pairs.Rdata	R Binary	A combined pairs table of final trades for all of the markets, appending together the pairs tables from each NAICS market
Pairs.csv	CSV Table	A combined pairs table of final trades for all of the markets, appending together the pairs tables from each NAICS market

3.5.6 COMPONENT 5: DAILY SAMPLE

Description

In this component, the transition is made to the regional touring model. The model takes a sample from the overall table of annual shipments for local Chicago, daily simulation.

Inputs and Parameters

Table 14 shows the input parameter that is used in the daily sample step of the model. There are no new inputs used in this model component.

TABLE 14: PARAMETERS USED IN THE DAILY SAMPLE STEP

PARAMETER	BASE VALUE	DESCRIPTION
annualfactor	310	Annualization factor to select a daily sample from the annual shipments

Model Steps

The daily sample component of the model applies the following steps:

1. A shipments database is created for just shipments to, from, or within the CMAP region.

2. Daily shipment frequency is calculated as the annual purchase quantity divided by the shipment weight, with the conversion to daily frequency made using the *annualization* parameter.
3. Not all of the pairs simulated in the PMGs have daily deliveriesand/or pickups. A random number between 0 and 1 is generated and firm pairs for which the random number is less than the daily frequency are assumed to have deliveries on the day being simulated.
4. Firm pairs with a daily frequency greater than one have daily deliveries and/or pickups.
5. From this point onwards only the selected firm pairs and corresponding shipments are simulated to represent a typical day scenario for deliveries and/or pickups in the CMAP region.

Outputs

No output tables are produced by this component. The shipments database (a data.table object) is passed directly to the next model component.

3.5.7 COMPONENT 6: WAREHOUSE ALLOCATION

Description

Once the daily sample has been taken, for those shipments that follow an indirect distribution channel via a distribution center, warehouse, or consolidation center the model assigns transfer points in the Chicago region.

Inputs and Parameters

There are no new inputs or parameters used in the warehouse allocation component of the model.

Model Steps

The warehouse allocation component of the model applies the following steps:

1. A local warehouse (within the CMAP modeling area) is randomly selected for indirect distribution channel shipments. This warehouse would represent the last transfer stop for incoming shipments and first transfer stop for outgoing shipments.
2. Based on the mesozone of the warehouse and whether or not a shipment is incoming or outgoing, the origin and destination mesozones of all the shipments that are to be simulated in a day are determined.
3. If the distribution channel is direct, the origin mesozone is the mesozone of the seller; if not, it is the warehouse mesozone (in the case of a drop-off a truck will start from the warehouse with a shipment and in the case of a pick-up, the truck will start from the warehouse without a shipment).
4. If the distribution channel is indirect and the buyer is outside the CMAP modeling area, the destination mesozone is the seller mesozone (to simulate a pick-up); if not, it is the buyer mesozone.

Outputs

No output tables are produced by this component. The shipments database (a data.table object) is passed directly to the next model component.

3.5.8 COMPONENT 7: VEHICLE CHOICE AND TOUR PATTERN

Description

This model component applies the vehicle choice and tour pattern model. This is a joint model that assigns each shipment to a particular size of vehicle and whether it will be picked up or delivered on a direct tour (i.e. one that includes only a single stop) or a peddling tour (one that includes multiple stops).

Inputs and Parameters

Table 15 shows the input tables that are used in the vehicle choice and tour pattern step of the model. The input tables are read into memory at the start of this step of the model and named as shown in the object name column of the table. Details of the layout of the input tables are described in Section 4.1. Table 16 shows the input parameters that are used in the vehicle choice and tour pattern step of the model.

TABLE 15: INPUTS TO THE VEHICLE CHOICE AND TOUR PATTERN STEP

FILENAME	OBJECT NAME	DESCRIPTION
data_emp_cbpzone.csv	emp_cbpzone	Number of employees by zone
data_mesozone_emprankings.csv	mzemp	Employment ranking by industry by county
model_vehicle_tourpattern.csv	vehtourpat	Vehicle and tour pattern model variables and coefficients

TABLE 16: PARAMETERS USED IN THE VEHICLE CHOICE AND TOUR PATTERN STEP

PARAMETER	BASE VALUE	DESCRIPTION
wgtmax_2axl	35000	Maximum load weight of light duty truck in pounds
wgtmax_3axl	65000	Maximum load weight of medium duty truck in pounds
wgtmax_semi	100000	Maximum load weight of heavy duty truck in pounds

Model Steps

The vehicle choice and tour pattern component of the model applies the following steps:

1. There are six alternatives for this model, which are combinations of two tour patterns (direct and multi-stop) and three vehicle types (2 axle, 3-4 axle, and semi/trailer). The model is influenced by commodity type, destination type, pick-up/drop-off weight and county employment in the destination zone.
2. Additional variables required by the model are created, including dummy variables indicating whether the shipment is a food product or manufacture product, whether the delivery is a pick-up or drop off, the log transformed shipment weight, and dummy variables for the industry of the business establishment where the delivery or pick-up will take place.
3. County employment data is merged with the dataset of daily shipments to be simulated.
4. The logit prediction function is applied to predict the tour pattern and vehicle type choice for each daily shipment.
5. Shipments using a direct distribution channel are forced to have a direct tour pattern keeping the vehicle type choice same as that output by the simulation/prediction.
6. For very large total daily delivery amounts (i.e., the sum of all shipments in the sample day from the supplier to the buyer), the weight of just the shipments could exceed the capacity of the vehicle

assigned by the mode. In those cases, the vehicle choice is moved up to the next vehicle size category that could successfully carry the load.

7. For even larger total daily delivery amounts, the weight of just the deliveries could exceed the capacity of any single truck to make in one delivery. In those cases, the shipments are split into separate shipments for individual delivery and are simulated separately in subsequent model steps.

Outputs

Table 17 below, show the outputs that are produced by the vehicle choice and tour pattern step of the model. Additional details such as the layout of output tables are presented in Section 4.3.

TABLE 17: MODEL OUTPUTS FROM THE VEHICLE CHOICE AND TOURS PATTERN STEP

FILENAME	TYPE OF OUTPUT	DESCRIPTION
vehtourpat_allcommodities.csv	CSV Table	Frequency of each alternative (combination of tour patterns and vehicle types) for all commodities
vehtourpat_food.csv	CSV Table	Frequency of each alternative (combination of tour patterns and vehicle types) for food commodity group
vehtourpat_mfg.csv	CSV Table	Frequency of each alternative (combination of tour patterns and vehicle types) for manufacturing commodity group

3.5.9 COMPONENT 8: STOP SEQUENCE

Description

In this component, additional detail is added to the structure of the delivery tours. The model applies the number of tours model, the clustering model, and the stop sequence model to identify the full set of stops that comprise each tour and the order in which they are made.

Inputs and Parameters

Table 18 shows the input tables that are used in the stop sequence step of the model. The input tables are read into memory at the start of this step of the model and named as shown in the object name column of the table. Details of the layout of the input tables are described in Section 4.1. There are no new parameters used in this model component.

TABLE 18: INPUTS TO THE STOP SEQUENCE STEP

FILENAME	OBJECT NAME	DESCRIPTION
data_mesozone_centroids.csv	mz_centroids	Zone centroid coordinates
data_mesozone_skims.csv	mz_skims	Skims for origins and destinations with available path costs and times
model_numberoftours.csv	numberoftours	Number of tours model variables and coefficients

Model Steps

The stop sequence component of the model applies the following steps to determine the number of truck tours for each shipment, cluster the indirect shipments according to the number of truck tours category, and sequencing all the stops within a tour:

1. There are three alternatives for the category of number of truck tours: all (stops) in one tour, two tours, three tours, and four tours. All in one tour means the shipment belongs to a tour category in which a truck covers all the stops assigned to it in a single tour. The purpose of this is to be able to determine the number of tours for all the stops serviced from a particular warehouse.
2. The additional required variables to apply the number of tours model are created. The model requires pick-up/drop-off weights, buyer and supplier industry categories, etc.
3. The model is applied to the daily shipments dataset.
4. The shipment stops are now clustered based on the number of truck tours category predicted. This is done only for indirect shipments since direct shipments are serviced by a direct tour with a single stop.
5. A separate vehicle type variable is created from the tour pattern vehicle type choice.
6. The count of shipment stops is obtained by warehouse, number of tours category, and vehicle type. Vehicle type is also used here in the aggregation so that stops grouped/clustered together have the same vehicle type since they are assumed to be serviced by the same vehicle.
7. It is ensured that the count of shipment stops is at least as much as the number of tours in the corresponding category. For example, if a particular shipment falls in a 4-tour category, there should be three more shipments in the 4-tour category that are assigned to the same warehouse for consistency. If that is not the case, the number of tours category is modified to be consistent.
8. The count of shipment stops is obtained again by warehouse, number of tours category (modified), and vehicle type.
9. The X and Y coordinates for each mesozone are merged into the shipments data by destination mesozone (stop mesozone).
10. A function clusters the shipments for each warehouse by the number of tours category and vehicle type. For example, if a warehouse has five indirect shipment stops assigned to it and all of them fall in the two tours category using 2-axle truck, the clustering function would cluster the five stops in two clusters using the Euclidean method.
11. After all the shipments stops are clustered into specific tours, unique tour and trip IDs are assigned to all the records.
12. For sequencing all the stops within a tour, a greedy algorithm is applied. It assigns the first stop as the one that is closest to the base (warehouse) based on the travel times in the mesozone skims. It

then keeps adding the next closest stop from the previous stop to the trip sequence until all of the stops are sequenced. After the final stop, the truck returns to the base as the last trip of the tour.

13. The truck load at each point during the tour is calculated based on the sequence of deliveries and the shipment weights.

Outputs

Table 19 below, show the outputs that are produced by stop sequence step of the model. Additional details such as the layout of output tables are presented in Section 4.3.

TABLE 19: MODEL OUTPUTS FROM THE STOP SEQUENCE STEP

FILENAME	TYPE OF OUTPUT	DESCRIPTION
stopseq_numstopspertour.csv	CSV Table	Frequency of each stop count category
stopseq_tourcatbyshipsize.csv	CSV Table	Frequency of tour category by shipment size
stopseq_stopcluster_sample.png	PNG File	Sample chart of stop clustering for each tour type category

3.5.10 COMPONENT 9: STOP DURATION

Description

This model component applies the stop duration model to estimate the time spent at each delivery or pick-up activity.

Inputs and Parameters

Table 20 shows the input table that is used in the stop duration step of the model. The input table is read into memory at the start of this step of the model and named as shown in the object name column of the table. Details of the layout of the input tables are described in Chapter 4. There are no new parameters used in this model component.

TABLE 20: INPUTS TO THE STOP DURATION STEP

FILENAME	OBJECT NAME	DESCRIPTION
model_stopduration.csv	stopduration	Stop duration model variables and coefficients

Model Steps

The stop duration component of the model applies the following steps:

1. There are six alternative stop durations in the stop durations model: less than or equal to 15 minutes, 15-30, 30-45, 45-60, 60-75, and greater than 75 minutes.
2. Additional variables required to apply the model are created in the shipments datasets.

3. The logit simulation function is used to predict the stop duration categories for both direct and indirect shipment datasets.
4. The exact stop duration in both datasets is assigned as the mid-point of the respective stop duration category:
    a. 0-15 min = 0.25 hrs.
    b. 15-30 min = 0.375 hrs.
    c. 30-45 min = 0.625 hrs.
    d. 45-60 min = 0.875 hrs.
    e. 60-75 min = 1.125 hrs.
    f. >75 min = 2 hrs.

Outputs

Table 21 below, show the outputs that are produced by stop duration step of the model. Additional details such as the layout of output tables are presented in Section 4.3.

TABLE 21: MODEL OUTPUTS FROM THE STOP DURATION STEP

FILENAME	TYPE OF OUTPUT	DESCRIPTION
stopdur_durationbytourtype.csv	CSV Table	Frequency of stop duration categories by tour types

3.5.11 COMPONENT 10: TOUR TIME OF DAY

Description

During this step, the model applies the time of day model, which estimates the times that tours begin. Each individual trip in the tour are then given a start and end time.

Inputs and Parameters

Table 22 shows the input table that is used in the tour time of day step of the model. The input table is read into memory at the start of this step of the model and named as shown in the object name column of the table. Details of the layout of the input tables are described in Chapter 4. There are no new parameters used in this model component.

TABLE 22: INPUTS TO THE TOUR TIME OF DAY STEP

FILENAME	OBJECT NAME	DESCRIPTION
model_timeofday.csv	tod	Time of day model variables and coefficients

Model Steps

The tour time of day component of the model applies the following steps:

1. The tour time of day multinomial logit model simulates the time of departure of the first trip in a particular truck tour. There are five alternatives for this – before 6 AM, 6-8 AM, 8-9 AM, 9-10 AM, and after 10 AM.
2. Additional variables required to apply the model, such as total tour and stop durations, are calculated for both direct and indirect tours.
3. Time of day categories are simulated for direct and indirect tour datasets. Based on the category of time of day an exact time is assigned to the tour start as follows:
    a. Before 6 AM = 5
    b. 6-8 AM = 7
    c. 8-9 AM = 8.5
    d. 9-10 AM = 9.5
    e. After 10 AM = 10.5
4. For indirect/multi-stop tours, the start times for each of the individual trips are calculated by adding the travel duration to the stop duration at the current stop to the start time of the previous trip. Since the tour start time or the start time for the first trip is already simulated, start times for all other trips in the tour can be calculated.
5. The final portion of this model component reviews the tours against two constraints and split tours where they do not meet the constraint:
    a. Vehicle loading: at no point during the tour should the load on the vehicle exceed the capacity of the vehicle. Tours are divided to meet the vehicle's weight capacity
    b. Tour start time: the threshold for the latest trip start time was set as 10 PM. Shipments flowing in and out of the CMAP modeling area had been randomly allocated to the warehouses in the area. In certain cases, due to the number of shipments to be delivered/picked up and/or due to the locations of shipments, it is found that certain tours had trips starting very late in the day. Since it is unlikely that such trips occur, these tours are broken down further to bring all the trip start times within a reasonable range.
6. The number of new tours into which these tours are broken down is determined by the greater of the number of 8-hours periods within the total tour time of the tours and the ratio of the maximum load on the vehicle during a tour to its capacity. For example, if the total tour time for a particular tour (which has a trip starting later than 10 PM) is 12 hours, it is split into two tours and a 19-hour tour is split into three tours. Similarly, a tour with a maximum load of 50,000lbs using a vehicle with a capacity of 35,000lbs will be split into two tours. The split tours could be interpreted as a new driver/vehicle added to the set of tours from a warehouse.
7. The vehicle type is kept the same as in the original/parent tour. All the stops are then clustered using the clustering algorithm into the number of sets that the tour is to be split into. Each cluster of stops is then sequenced using the greedy algorithm. The tour start times of the split tours are kept the same as that of the original tour. The individual trip start times are recalculated to reflect the new split tour start times.
8. Steps 5 through 7 are repeated until all trips start before 10 PM and at all points on all tours, vehicle are loaded at or below the vehicle capacity.

Outputs

Table 23 below, show the outputs that are produced by tour time of day step of the model. Additional details such as the layout of output tables are presented in Section 4.3.

TABLE 23: MODEL OUTPUTS FROM THE TOUR TIME OF DAY STEP

FILENAME	TYPE OF OUTPUT	DESCRIPTION
tod_todbytourtype.csv	CSV Table	Frequency of time of day categories by tour types

3.5.12 COMPONENT 11: PREPARE TRIP TABLES

Description

This final component of the model aggregates the individual trips that form the pick-up and delivery tours into zone-to-zone trip tables by time period.

Inputs and Parameters

There are no new inputs or parameters used in the prepare trip tables component of the model.

Model Steps

The prepare trip tables component of the model applies the following steps:

1. Text descriptions of vehicle types and times of day are added to the direct and indirect tour datasets
2. The direct and indirect tour datasets are combined together into a single tours dataset.
3. The tours dataset is aggregated by origin mesozone, destination mesozone, vehicle type, and time of day to produce the final trip tables.

Outputs

Table 24 below, show the outputs that are produced by prepare trip table step of the model. Additional details such as the layout of output tables are presented in Section 4.3.

TABLE 24: MODEL OUTPUTS FROM THE PREPARE TRIP TABLE STEP

FILENAME	TYPE OF OUTPUT	DESCRIPTION
trip_table.csv	CSV Table	Number of trips by origin mesozone, destination mesozone, vehicle type, and time of day

3.6 PROCUREMENT MARKET GAME (PMG)

This section describes the PMG that underlies the buyer-seller matching model. PMG is a standalone executable program, written in C++, which implements an agent-based model of a procurement market. It operates as a kernel within the meso-scale freight model, receiving inputs from the meso-scale model and producing a set of matched buyer-seller pairs with transactional information as outputs.

The first sub-section, 3.6.1, discusses the development history of PMG.exe. Sub-section 3.6.2 provides background information on the theory behind PMG, which is important to understanding its behavior. Sub-section 3.6.3 is a narrative of game play, what happens at each step in summary form. Sub-section 3.6.4 describes the meso-scale freight model inputs to PMG with details on buying and selling agents, and their transaction costs. A description of PMG results files may be found in Sub-section 3.6.5.

An example of PMG configuration settings is provided in Sub-section 3.6.6, and Sub-section 3.6.7 describes how to run PMG from a command line. Sub-section 3.6.9 is a detailed description of the algorithm of the PMG application program, including the fundamental steps and embedded agent logic. Sub-section 3.6.10 describes how payoffs are calculated, variations on payoff structures and their implications, and the role of feedback (agent learning).

3.6.1 DEVELOPMENT HISTORY

The creation of PMG started with an open-source program called "TNG" (trade network game), originally developed by an agent-based computational economics research group at Iowa State University (McFadzean et al 2001). TNG was developed as a research tool to demonstrate principals of game theory for pedagogical purposes. Accordingly, TNG is structured to be flexible and general in its implementation of games, using hypothetical payoff values for a small, finite set of undifferentiated anonymous agents.

To create PMG, the original TNG code was modified and extended in numerous places, principally to achieve the following aims:

- Differentiation between agent types and preferences;
- Specify algorithmic logic relevant to supply chain relationship functions in industrial procurement markets;
- Enable input and use of real-world data in the calculation of multi-variable payoff functions; and
- Optimize code by eliminating unneeded TNG functions and maximizing processing speed to handle billions of agent interaction events.

3.6.2 THEORETICAL BACKGROUND

The modeling approach used in PMG is inspired by the theory of buyer-seller networks proposed by Kranton et al (2001) and incorporates concepts for agent interactions from other sources. Cachon (2003) provides a detailed mathematical treatment of various contexts, assumptions and incentives for buyer-seller cooperation in supply chain contracting, some of which PMG can approximate under certain parameter assumptions. Chen (2003) describes the principal-agent games that arise in supply chain trading decisions and which lie at the heart of agent interactions in PMG.

The PMG buyer-seller matching model is formulated as a Bayesian principal-agent game with asymmetric information and belief updating. Agents do not know the nature of the other agents prior to game and learn about the other agents through iterative rounds of play. In principal-agent games (Laffont et al 2002), players take turns. The principal is the less-informed player, while the agent is assumed to be more informed. Game types vary depending on which agent moves first. Both players are assumed to attempt to maximize profits, or utility, and form expectations based on asserted payoff values for different outcomes.

Games in which the more informed player moves first are called signaling games (Stigligtz et al 1990; Chen 2003; Rasmusen 2006a; Kübler et al 2008). For example, a supplier makes the first move and offers to supply a manufacturer with widgets. The supplier knows about the quality of its widgets and service capabilities and signals this to the potential buyer through price. The manufacturer then decides whether to accept the offer, risking that what the supplier provides may be of lesser quality. The true value of the supplier's product/efforts is revealed to the buyer upon delivery.

Games in which the principal, the less-informed player, moves first are often referred to as screening games (Stigligtz et al 1990; Chen 2003; Rasmusen 2006a; Kübler et al 2008). For example, the manufacturer would be able to review the published offerings of potential suppliers and would initiate interaction with a supplier through the offer of a contract. The seller then decides whether to accept the contract or reject it. If the seller accepts the contract, it reveals the true value to the seller through its efforts.

In both signaling and screening games, the quality of the agent's product or efforts are assumed to be determined by "nature" prior to the game's beginning, but are unknown to the principal. That is, the agent is of a certain type, a high or low performer/quality. For a single pair of agents, equilibrium solutions may be derived through linear programming methods that derive the optimal price for the principal to offer/accept in order to incentivize the desired behavior (service) from the agent.

Games in which the value of the outcome of the agent's efforts is determined post-contract acceptance are commonly referred to as games of moral hazards (Rasmusen 2006b). Moral hazards include relationships such as between insurance companies and policy holders, landlords and tenants, shareholders and company presidents, and similar relationships in which behavior or other circumstances are unknown prior to and determined after contract acceptance These games are said to have post-contract hidden information, with expected outcomes subject to probability distributions. For a set of one principal and one agent, they can be solved using dynamic programming methods.

3.6.3 SUMMARY OF PROCUREMENT MARKET GAME (PMG) PLAY

PMG is similar to a screening game of moral hazards with post-contractual hidden information, but is considerably more complex because it involves multiple principals and agents, called buyers and sellers, and is a repeated game with feedback and learning. Figure 2, below, shows the extensive form of the game between a single buyer and seller.

The game begins with an initial market (M) assessment by a buyer (B), who needs to procure some amount of an input commodity to meet its annual production demands. The buyer evaluates whether a prospective seller would provide a relatively high or low expected utility if given a contract. Depending on information assumptions, and how much learning has taken place in previous encounters, a buyer will offer a contract to a seller with the highest expected utility among the available sellers. In the initial iteration of the game, these expected values are naïve.

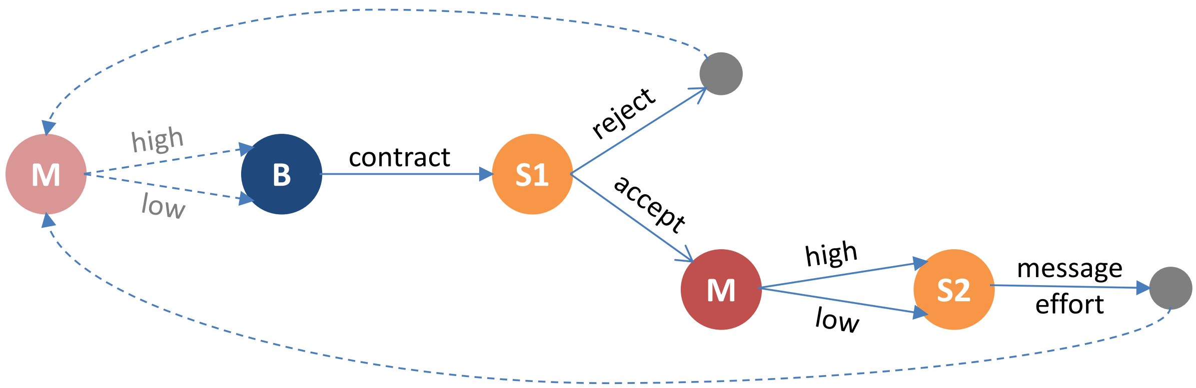

FIGURE 2: PROCUREMENT MARKET GAME FOR A SINGLE BUYER-SELLER PAIR

The seller then moves next (S1), deciding whether to accept or reject the offered contract. Sellers order contracts in terms of expected revenues, which are a function of the amount each buyer is asking for in the contract and the price (which is determined by the seller).

The seller would like to capture as much revenue as possible; therefore, it will accept the contract offered by the buyer, unless it has already reached its production capacity through contracts to other buyers. Once sellers receive all of the offers from buyers, they order their contracts in descending value and accept all contracts up to their capacity limits. Any contract that cannot be filled is rejected.

The seller then fulfills the terms of the contract through its efforts (S2). Symbolically, we say that it passes a message back to the buyer. If the seller has sufficient capacity, the message will be that the order is filled in full (high). If the seller's capacity has been reached but there is room for partial fulfillment for one more buyer, the seller will accept the contract but send a message back to the buyer indicating the proportion of the demand that has been satisfied (low).

PMG is a Bayesian game in the sense that players do not know the other player's type and utility preferences. Specifically, buyers do not know how many or the size of other contract offers an individual seller might have. If the seller has many competing contracts, it might not be able to fulfill the buyer's order after accepting the contract. Similarly, sellers do not know how many or the value of other contracts that the buyer may have offered to other sellers. A buyer may offer to procure a larger or smaller amount from the same seller in different iterations, depending on where that seller ranks in the buyer's evolving esteem. Since a player does not know how he/she rates in the mind of the other agent, the incentive to cooperate or defect is based on expected values from past encounters. As an agent accumulates more of these experiences through iterative games, the agent begins to rate higher the agents that are most likely to be stable trading partners, even if they were not initially their first choice. Nor do buyers know whether a seller may provide fractional fulfillment or reject their offer entirely. In fact, the seller will accept the contract and after the fact send information back to the seller about partial fulfillment.

The desire to calculate expected utilities and revenues as a function of realistic input variables led to the development of payoff values formulated as weighting factors. The input variables—quantities of commodities to be purchased, seller capacities to fill orders, prospective shipping times and unit costs—are evaluated by buying agents, and the expected revenues for these transactions are evaluated by selling agents, processes which are described in more detail in Section 3.6.4. Sub-section 3.6.10, below, describes how weighting factors are calculated and applied by agents in rating their experiences with other agents. These weighting factors upgrade or downgrade the nominal value of the trades and are used to update agent beliefs about other agents with whom they have traded in the past. Through iterative game play, agents will form attachments to particular trading partners, leading to stable alliances over a sufficient number of rounds.

3.6.4 MESO-SCALE FREIGHT MODEL INPUTS TO PMG

The meso-scale freight model produces three primary inputs to the PMG:

- Buyer agents
- Seller agents
- Costs for feasible combinations of buyers and sellers

Buyer Agents

The meso-scale freight model determines the annual production requirements for each synthetic firm in the model system. Using the BEA IO tables, the model creates for each producing firm a list of input commodity purchase requirements needed to satisfy production quantities. For all of the commodities represented in the 6-digit NAICS code (See Table 29), the meso-scale model identifies all of the firms seeking to buy that commodity and creates a pool of buyer agents who will participate in a sourcing market for that commodity. The default name for this file is PMG.buy.csv; however, it is possible to specify other file names by providing specific file names in the command line options for executing PMG (see Sub-section 3.6.7). Buyer agents are represented as records in a table with attributes defined as in Table 25, below, is a sample from a Buyer agent file. The first several fields in Table 25 are self-evident from the field descriptions; however, PurchaseAmountTons, PrefWeight and MaxFrac fields require further explanation.

Purchase amounts are a function of firm size and the direct requirements proportions from the IO tables. The meso-scale model simulates production quantities of each commodity in direct proportion to firm size as a share of total domestic production. Because the firm synthesis module creates firms in employee-size bins, this leads to a distribution of firms in any given industry that all have the same number of employees within the same bin. Accordingly, these firms will then all produce the same annual quantity of a commodity. Since the IO tables stipulate how much of an input commodity is required to produce a unit of the output commodity, firms of the same industry and employee size, will also seek to purchase the same amount of a particular input commodity. In most industries, the employee-size distribution is such that there are many firms in the smaller firms and fewer very large firms. This means that in most markets there are many small firms seeking to purchase the same small amount of an input commodity and, at the other end of the distribution, a few very large firms seeking to purchase the same large amount of that same input commodity. Inspection of Figure 3 illustrates this point. Since this "lumpiness" is rooted in the firm size distributions used to synthesize firms, it would be easy to change by converting the discrete employee-size bins into a continuous distribution, which would create similar variation in production and consumption quantities across firms.

Preference weights are utility function coefficients, which are intended to represent tradeoffs between attributes of the bundle of product and service offerings of potential sellers. In the initial implementation of

PMG, this is limited to two attributes, shipping time and unit cost. In theory, preference weights would be estimated from observed behavioral data, which were not available for this implementation. Therefore, the meso-scale model asserts proportional weights, based on a literature review of supply chain sourcing preferences for responsiveness (minimize shipping time) versus cost savings. Weights were assigned using the function described in Section 3.3.4 and reflect the commodity being produced, the commodity being purchased, and firm size. The primary considerations are whether the commodity to be purchased is of low or high value-to-weight ratio and whether it is perishable or time-sensitive in the context of the commodity being produced (functional or innovative).

Although the PrefWeight values shown in Figure 3 are positive-valued, they are transformed in PMG's internal utility calculations as costs, thereby representing disutility, as in the following equation in which β_t and β_c represent the weights on shipping time and unit cost, respectively, for the calculation of the utility accruing to Buyer *b* of a contract with Seller *s*.

$$Utility_{s \to b} = -\beta_t ShippingTime - \beta_c UnitCost$$

TABLE 25: BUYERS FILE FIELD DEFINITIONS

FIELD	DESCRIPTION
InputCommodity	NAICS code for commodity to be procured (consumed)
BuyerID	Buyer ID number in population of synthetic firms
Zone	Zone ID number in model database (synthetic firm's geographic location)
NAICS	NAICS code for synthetic firm
Size	Number of employees in firm
OutputCommodity	NAICS code for commodity to being produced by this firm (usually but not always same as firm's NAICS code)
PurchaseAmountTons	Total annual amount needed to purchase (pounds)
PrefWeight1_UnitCost	Proportional utility weight on Attribute1_UnitCost in Costs file (producer cost plus transport and logistics costs, normalized by quantity)
PrefWeight2_ShipTime	Proportional utility weight on shipping time (days in the Cost file)
SingleSourceMaxFraction	Maximum proportion of purchase quantity from a single seller agent; valid range: 0 < x <= 1.0; default is 1.0

InputCommodity	BuyerID	Zone	NAICS	Size	OutputCommodity	PurchaseAmountTons	PrefWeight1_UnitCost	PrefWeight2_ShipTime	SingleSourceMaxFraction
113000	7387425	274	113000	0	113000	399315.2761	0.8	0.2	0.8
113000	7387431	274	322120	0	322120	283905.4847	0.8	0.2	0.8
113000	7387437	274	33712A	0	33712A	7936.862771	0.8	0.2	0.8
113000	7387443	276	321910	0	321910	988723.5806	0.8	0.2	0.8
113000	7387449	276	325320	0	325320	75348.82306	0.8	0.2	0.8
113000	7387455	278	113000	0	113000	922701.362	0.8	0.2	0.8
113000	7387461	278	322120	0	322120	656022.9299	0.8	0.2	0.8
113000	7387467	278	33712A	0	33712A	18339.77943	0.8	0.2	0.8
113000	7387473	281	321910	0	321910	8623641.631	0.8	0.2	0.8
113000	7387479	281	325320	0	325320	657192.0202	0.8	0.2	0.8
113000	7387485	283	113000	0	113000	211619.8986	0.8	0.2	0.8
113000	7387491	283	322120	0	322120	150457.6796	0.8	0.2	0.8
113000	7387497	283	33712A	0	33712A	4206.195442	0.8	0.2	0.8
113000	7387503	284	321910	0	321910	9424916.089	0.8	0.2	0.8
113000	7387509	284	325320	0	325320	718255.6871	0.8	0.2	0.8
113000	7387515	285	113000	0	113000	6062917.713	0.8	0.2	0.8
113000	7387521	285	322120	0	322120	4310617.937	0.8	0.2	0.8
113000	7387527	285	33712A	0	33712A	120507.6509	0.8	0.2	0.8
113000	7387533	287	321910	0	321910	13231955.62	0.8	0.2	0.8
113000	7387539	287	325320	0	325320	1008383.235	0.8	0.2	0.8

FIGURE 3: BUYER INPUT FILE EXCERPT FOR A SINGLE MARKET (COMMODITY #113000)

In the absence of estimated parameters, an intuitive alternative is to represent proportional weights on the utility of shipping time and cost; however, to do so it is necessary to account for the differences in values of the variables themselves. Using all potential buyer-seller agent pairs in the commodity market (described below under Costs File), the idea is to create weights that, when applied to the average values for each variable, produce utilities for each variable that are proportional to the expected proportional contributions of each variable to the utility calculation.

For example, if the expected proportional weights on shipping time and cost were 0.60 and 0.40, respectively, calculation of the rescaled utility weights would be as follows:

Average market shipping time *t*:	2.03 days	
Average market unit cost *c*:	\$0.78 per ton	
Un-weighted proportions:	$\rho_t^0 = \frac{2.03}{(2.03+0.78)} = 0.72$	$\rho_c^0 = \frac{0.78}{(2.03+0.78)} = 0.28$
Expected proportional utility:	$\rho_t^* = 0.60$	$\rho_c^* = 0.40$
Adjusted weights:	$\beta_t = \frac{0.60}{0.72} = 0.83$	$\beta_c = \frac{0.40}{0.28} = 1.45$
Utility of average shipping time:	$U_t = -0.83 * 2.03 = -1.68$	
Utility of average unit cost:	$U_c = -1.45 * 0.78 = -1.12$	
Utility proportions:	$\rho_t^u = \frac{-1.68}{(-1.68-1.12)} = 0.60$	$\rho_c^u = \frac{-1.12}{(-1.68-1.12)} = 0.40$

For this example market, the adjusted weights calculated for β_t and β_c would be the values appearing under PrefWeight1_UnitCost and PrefWeight2_ShipTime in the Buyer file. Note that in Figure 3 the 0.80 and 0.20 values shown under these headings were not the result of these calculated adjustments.

The meso-scale model currently assumes that all buyers in the commodity market have the same preference weights for the commodity contract. Alternatively, it would be possible to calculate different proportional weights for different types of buyers within the same commodity market. Automation of this calculation has not been implemented in the meso-scale model thus far, but it would be possible to perform the above calculations for groups of buyers of the same type manually.

Maximum fractions constraints are used to specify constraints on the proportion of a buying agents purchase quota that can be contracted to a single seller. This is an optional user-configurable setting that it intended to represent risk hedging, which is commonly practiced by supply chain managers for high-value and time-sensitive commodities. For example, if SingleSourceMaxFraction were set to 0.80, then a buyer seeking to purchase 60 tons of a bulk material could purchase at most 48 tons from a single seller and would have to find at least one other seller from which to procure the remaining 12 tons. As another example, if SingleSourceMaxFraction were set to 0.40, then a buyer seeking to purchase 60 tons of a bulk material would have to procure this commodity from at least three sources since no single source could provide more than 24 tons. Relaxation of these constraints for specific buyer agents will happen automatically within the PMG logic, as described below in the algorithm description (Sub-section 3.6.9), if conditions warrant it.

The performance implications of maximum fraction constraints are to make procurement more difficult in markets where there are many buyers and few sellers, which could lead to excessive run times. Thus, they should be used with caution, particularly in markets where the buyer-to-seller agent ratio is 1000:1 or greater.

Seller Agents

For all of the commodities represented in the 6-digit NAICS code (See Table 29), the meso-scale model identifies all of the firms that produce that commodity and creates a pool of seller agents, who will participate in a sourcing market for that commodity. It is important to remember that the same firms that produce commodities also consume commodities. In most cases, each firm will produce commodities for a single commodity code, but will purchase inputs to their production process from many other commodity codes. Each synthetic firm will be represented in the model system by, in most cases, a single selling agent for the commodity that they produce, and by many buying agents, each responsible for purchasing a different input commodity. Thus, each firm will be involved in multiple PMG simulations.

It is also possible that a firm produces and consumes commodities classified in the same commodity code, particularly for electronics and similar high-technology items in which high-tech components (e.g., micro-chips) are need to produce other high-tech devices.[4] In such markets, a firm may have one buying agent and one selling agent in the same market game. Each agent will function as either a pure buyer or a pure seller; there are no resellers (dealers) in PMG. It is possible, however, for a firm's buying agent to transact with its selling agent, which may be considered an instance of vertical integration. Vertical integration has not been made an explicit strategy within PMG, but can result implicitly in markets such as describe here. Explicit vertical integration strategies could be developed further in future versions of PMG.

The default name for this file is PMG.sell.csv; however, it is possible to specify other file names by providing specific file names in the command line options for executing PMG (see Sub-section 3.6.7). Seller agents are represented as records in a table with attributes defined as in

Table 26, below, is a sample from a Seller agent file. It should be relatively easy to follow; however, OutputCapacityTons and NonTransportUnitCost merit additional clarification.

TABLE 26: SELLER FILE FIELD DEFINITIONS

FIELD	DESCRIPTION
OutputCommodity	NAICS code for commodity produced and sold by this firm (seller)
SellerID	Seller ID number in population of synthetic firms
Zone	Zone ID number in model database (synthetic firm's geographic location)
NAICS	NAICS code for synthetic firm
Size	Number of employees in firm
OutputCapacityTons	Synthetic firm's annual total production/sales capacity in pounds for this commodity code
NonTransportUnitCost	Production cost in dollars per pound for this firm and commodity code (does not include transport and logistics costs)

Output capacities are proportional to firm size. Thus, seller agents are subject to the same "lumpiness" as buyers in terms of production quantities. For each market to be simulated using PMG, it is important to check whether the sum of the output capacity of all seller agents is sufficient to meet the combined demand

[4] With 6-digit NAICS specificity, this will be rare.

of all buyer agents. Buyers will purchase many different input commodities in order to produce a single output commodity; therefore, each firm is represented many more times as a buyer than it is a seller. Markets in which the buyer-to-seller agent ratio is 1000:1 or greater are common; however, even with such large ratios the total seller capacity should be greater than the total buyer demand. A market in which there is insufficient seller capacity will have trouble converging, leaving many buyers with nothing. Capacity shortfalls are most likely to occur when doing batch processing to save computational time. To safeguard against this, the meso-scale model actually forces total capacity to be greater than demand: purchase requirement are factored down until lower the ratio of production capacity to purchase requirements until it is greater than 1.1.

Non-transport unit costs represent producer prices, excluding transport and logistics costs. The meso-scale freight model derives these costs from SCTG groupings of the FAF3 commodity prices. This requires a mapping from the 260 6-digit NAICS commodity codes used in PMG that describe industries which make commodity to the 43 SCTG codes, which results in a somewhat coarser set of commodity prices. The commodity sales prices for domestic wholesaler offerings are the SCTG-derived producer price, plus a markup value of 20%. The commodity sales prices for foreign sellers are the SCTG-derived producer prices, plus a discount factor of 10%. For example, in Figure 4, the sellers with the smaller ID numbers, 1-61, represent foreign firms, and their non-transport unit costs are 90% of the firms with the 7-digit ID numbers. These price factors, wholesalemarkup and foreignprodcostfactor, may be configured by the user in the file *scenario_variables.R* as described in 4.2. More precise representation of producer prices for various commodities could be developed if the data were available.

OutputCommodity	SellerID	Zone	NAICS	Size	OutputCapacityTons	NonTransportUnitCost
113000	7373499	275	113000	0	524376.0205	0.0192915
113000	7373545	389	113000	0	61944.37965	0.0192915
113000	7373566	448	113000	0	1232045.201	0.0192915
113000	7373585	485	113000	0	1896171.889	0.0192915
113000	7373520	334	113000	1	9028173.03	0.0192915
113000	7373589	313	113000	1	8915169.893	0.0192915
113000	7373554	419	113000	2	16175984.24	0.0192915
113000	7373515	323	113000	3	28736697.51	0.0192915
113000	7373572	456	113000	4	31822823.52	0.0192915
113000	1	151	113000	10	84879224.54	0.021435
113000	7	153	113000	10	84879224.54	0.021435
113000	13	153	113000	10	84879224.54	0.021435
113000	19	153	113000	10	84879224.54	0.021435
113000	25	153	113000	10	84879224.54	0.021435
113000	31	152	113000	10	84879224.54	0.021435
113000	37	158	113000	10	84879224.54	0.021435
113000	43	158	113000	10	84879224.54	0.021435
113000	49	163	113000	10	84879224.54	0.021435
113000	55	175	113000	10	84879224.54	0.021435
113000	61	175	113000	10	84879224.54	0.021435

FIGURE 4: SELLER INPUT FILE EXCERPT FOR A SINGLE MARKET (COMMODITY #113000)

Costs File

The meso-scale model creates a cost file for each market game, using a pool of buyer and seller agents that will participate in the game. The Costs file provides records for all feasible combinations of matches between buyers and sellers within the same market. The default name for this file is PMG.costs.csv; however, it is possible to specify other file names by providing specific file names in the command line options for executing PMG (see Sub-section 3.6.7). Definitions for the fields found in this file are shown in

Table 27, and a sample excerpt from a Cost file is shown in Figure 5. The meso-scale model calculates and appends transaction attributes to each record in the file.

TABLE 27: COSTS FILE FIELD DEFINITIONS

FIELD	DESCRIPTION
SellerID	Seller ID number in population of synthetic firms
BuyerID	Buyer ID number in population of synthetic firms
Attribute2_ShipTime	Expected shipping time in days for each shipment from this supplier to this buyer
Attribute1_UnitCost	Producer cost plus transport and logistics costs, normalized by quantity

SellerID	BuyerID	Attribute2_ShipTime	Attribute1_UnitCost
1	894987	0.495841643	85.89256843
1015	894987	0.495841643	85.89256843
1021	894987	0.495841643	85.89256843
1039	894987	0.495841643	85.89256843
1051	894987	0.495841643	85.89256843
1003	897028	0.495841643	85.89256843
1009	897028	0.495841643	85.89256843
1015	897028	0.495841643	85.89256843
1039	897028	0.495841643	85.89256843
1063	897028	0.495841643	85.89256843
1027	897028	0.495841643	85.89256843
1003	894987	0.495841643	85.89256843
1009	894987	0.495841643	85.89256843
1027	894987	0.495841643	85.89256843
1033	894987	0.495841643	85.89256843
1045	894987	0.495841643	85.89256843
1057	894987	0.495841643	85.89256843
1063	894987	0.495841643	85.89256843
6381777	894987	0.495841643	85.89256843
1	897028	0.495841643	85.89256843

FIGURE 5: COSTS INPUT FILE EXCERPT FOR A SINGLE MARKET (COMMODITY #113000)

Shipping times are tabulated from the mode-path choice model outcomes, described in Section 3.4. These represent total shipping times, which include assumptions about transshipment node times.

Unit costs are calculated as the sum of the non-transport unit costs from the Sellers file and the transport-logistics cost function described in Section 3.3, normalized on a per unit basis. It is important to remember that the transport and logistics costs embedded in this calculation take into account annualized logistics costs, such as ordering, storage, damage, inventory-in-transit, transport-handling, carrying, and safety stock costs.

Thus, every feasible buyer-seller agent pair represents a contract to deliver the amount of a good demanded by the buyer agent for an assumed cost per unit and average shipping time per shipment. Here, feasibility refers to mode-path connections. For example, a truck-rail-truck model path is not available for imports from China.

As described below in Sub-section 3.6.9, buyer agents choose seller agents based on differences in utilities, which are calculated as a weighted combination of the attributes shown in

Table 27 and describe above under Buyer Agents. To facilitate a market game in which buyer agents with varying quantity requests are subject to the same utility preference weights, total costs are normalized by the quantity demanded, yielding a unit cost. Thus, buyers discriminate among seller bids based on differences in cost per unit ordered.

3.6.5 PMG OUTPUTS

When PMG runs, it produces an output file for each market game. The meso-scale model is configured to sub-divide NAICS commodity sourcing markets into processing batches in order to reduce combinatorics and save computational time, a procedure described in Section 3.5.3. Once all of the games have been run, the meso-scale model collates the results of individual batches and creates a single output file for each NAICS commodity market. The default name for this file is PMG.out.csv; however, it is possible to specify other file names by providing specific file names in the command line options for executing PMG (see Sub-section 3.6.7). Output-file field definitions are shown below in Table 87, and an example excerpt from an output file is shown below in Figure 6.

Each record in the table shows the trades made between buyer-seller pairs for which there was at least one trade contracted during at least one game iteration. Buyer-seller pairs that never trade are filtered out of the final output files to reduce storage space. As described in Sub-sections 3.6.6 and 3.6.9, PMG is typically run for multiple iterations to simulate agent learning through repeated trade encounters with other agents.

The quantity traded in the output file represents the sum of all trades made between a buyer-seller pair, expressed in pounds, and the number of trades that took place, summed over all iterations, is useful as a diagnostic metric to judge the stability of an outcome. By dividing the quantity traded by the number of trades, one could derive an average quantity traded or expected value. Those trading pairs that trade for the majority of iterations are likely to be stronger partners.

The last iteration quantity traded by the pair reflects the results of the last iteration and is the value that is used in the meso-scale model to represent annual shipments between pairs of buying and selling firms. This will not always be the partners who traded for the most iterations, particularly for small iterations. Depending on how the game is configured, it is possible that some experimentation or oscillation could take place during late rounds, particularly with large numbers of buyers and sellers.

BuyerId	SellerId	Quantity.Traded	Number.of.Trades	Last.Iteration.Quantity
7387425	841	79863	1	0
7387425	865	79863	1	0
7387425	925	319452	1	0
7387425	1021	79863	1	79863
7387425	1309	319452	1	0
7387425	1735	399315	2	319452
7387425	2089	79863	1	0
7387425	2389	319452	1	0
7387425	7381	319452	1	0
7387425	6381759	319452	1	0
7387425	6381855	79863	1	0
7387431	1	227124	1	227124
7387431	949	56781	1	56781
7387431	1771	56781	1	0
7387431	2023	227124	1	0
7387431	2119	56781	1	0
7387431	2671	56781	1	0
7387431	3091	227124	1	0
7387431	7189	227124	1	0
7387431	7411	56781	1	0

FIGURE 6: OUTPUT FILE EXCERPT

3.6.6 CONFIGURING GAME PLAY

PMG may be configured to represent different assumptions about producer and consumer (buyer and seller) behavior. PMG.exe is initialized using a text file with a (*.ini) extension. The default name for this file is PMG.ini; however, it is possible to specify other file names by providing specific file names in the command line options for executing PMG (see Sub-section 3.6.7). Figure 7, below, shows an example of a configuration file.

```
//
// pmg.ini
//
[pmg]

RandomSeed = 41

// number of iterations
IMax = 20

// want lots of detail about tradebots?
Verbose = 0

// recalculate alternate payoffs every iteration based on updated expected
payoffs
DynamicAlternatePayoffs = 1

// should initial expected tradeoffs know size of other traders?
ClairvoyantInitialExpectedPayoffs = 0

// should sellers accept offers based on order size instead of expected
payoff?
SellersRankOffersByOrderSize = 1

// multiplier to goose initial expected tradeoff to encourage
experimentation with other traders
InitExpPayoff = +1.2

Temptation = 1.5
BothCoop = 1
BothDefect = 0.8
Sucker = 0.67

// amount to downgrade expected payoff of seller who outright refuses a
trade offer by buyer
RefusalPayoff = 0.5

// negative payoff to sellers for not participating
WallflowerPayoff = 0

// buyers dont try to trade with sold out sellers
BuyersIgnoreSoldOutSellers = 1

// ratio at which buyers don't try to trade with sold out sellers
IgnoreSoldOutSellersMinBuyerSellerRatio = 100

// faster reading of input files but does less checks (ok for use with R)
RawFastParser = 1
```

FIGURE 7: AN EXAMPLE OF A PMG.INI INPUT FILE USED TO CONFIGURE THE GAMES

Operational notes on the parameter settings include the following:

- The role of the RandomSeed is to ensure repeatability. Random numbers in PMG are used only in a "quicksort" method, which randomly orders agents at the beginning of game play and determines the order in which buying agents screen selling agents. Algorithmic rules determine game outcomes, as described below in Sub-section 3.6.9.
- Parameters that provide binary options should be set to 1 (true) to invoke the option, or 0 (false, default) to turn them off. This pertains to: Verbose, DynamicAlternatePayoffs, ClairvoyantInitialExpectedPayoffs, SellersRankOffersByOrderSize, BuyersIgnoreSoldOutSellers , and RawFastParser.
- Other parameters require the input of real-number values, and if real numbers are not supplied, default values of 0 will be used, which will result in a nonsensical outcome.
- If RawFastParser=1, PMG will parse input files using a streamlined method that has fewer error checks. This has been tested with CSV files generated by R and found to be safe.

The comments in the sample file shown in Figure 7 provide a general indication of how each configuration is used; however, better understanding of how these parameter settings affect PMG, it is necessary to describe them in the context of game play. Sub-section 3.6.9, below, which describes the algorithm underlying PMG, and Sub-section 3.6.10, which describes payoff value calculations, provide this intuition.

By default, PMG will provide only status information on the screen when run from a command line, and nothing at all when run from within the meso-scale model. The Verbose setting, if set to 1 (true), will provide very detailed output on agent interactions for every iteration of the game. As described below in Sub-sections 3.6.7 and 3.6.8, this could be output to a text-format log file for more detailed study of interactions between agents, who are referred to as "tradebots" in the PMG source code.

3.6.7 RUNNING PMG

The meso-scale model runs PMG for every NAICS commodity market in the simulation, automatically creating Buyer, Seller, and Cost file inputs, collating Output files, and assigning file names based on commodity codes. When the meso-scale model initiates a run, it generates an operating system command line that includes arguments for all of the file names and paths needed to run PMG and keep track of commodity markets. It is also possible to run PMG in standalone mode from a command line prompt using these same arguments, using Buyer, Seller and Cost input files that have been created beforehand. This permits experimentation on individual market segments or perhaps toy data sets without the overhead of running the full meso-scale model.

The default values for running PMG at a command line assume "PMG" as the main portion of each input and output file name, followed by suffixes indicating the file type. For example, if all of the input files are located in the same directory as the PMG executable, then one could open a command prompt window and execute PMG by simply typing "PMG." The program would expect to find the following files:

PMG.ini

PMG.buy.csv

PMG.sell.csv

PMG.cost.csv

The default output file would be:

PMG.out.csv

To specify different file names and data paths, four command line options are available, which use single-letter designations preceded by a dash, as illustrated in the following examples:

1. **Specify ini file path:**

   -i C:\cmap\freight_model\PMG\pmg_payoff_scheme3.ini

   (If not specified, defaults to expecting to find it in the data directory)

2. **Specify data input and output file name prefixes:**

   -p PMG_run6

   (If not specified, defaults to PMG)

3. **Specify data directory path for input files and output files:**

   -d C:\cmap\freight_model\data

   (If not specified, defaults to looking for data in current working directory)

   Assuming the prefix specified in 2 above, this example would look for input data in (e.g.) C:\cmap\freight_model\data\PMG_run6.buy.csv (and PMG_run6. sell.csv and PMG_run6.costs.csv)

4. **Specify directory path for output file:**

   -o C:\cmap\freight_model\data \out

   (If not specified, defaults to putting output in data directory with input files)

   Puts output in C:\cmap\freight_model\data \out\PMG_run6.out.csv

These command line options may be combined all on a single (wrapped) line as follows:

```
C:\Projects\CMAP_Freight\_Software\PMG\PayoffTests\pmg.exe -i
C:\cmap\freight_model\PMG\pmg_payoff_scheme3.ini -p PMG_run6 -d
C:\cmap\freight_model\data -o C:\cmap\freight_model\data\out
```

For convenience, it is recommended that these command line options be saved in a text file with a *.cmd file extension, which can be run from a command prompt as follows:

```
c:\>runpmg.cmd
```

It is also possible, and recommended to send console messages to log file, which is particularly useful for runs with several of agents, or if Verbose=1. This may be done by typing a “>” followed by a file name after the command file, such as:

```
c:\>runpmg.cmd > run6.log
```

3.6.8 CONSOLE OUTPUTS AND LOG FILES

As shown in Figure 8, the verbose output format first generates a summary of PMG input values, as specified in the .ini file. Next, PMG outputs an initial summary of preference weights, attribute values, and utility values for each combination of buyer and seller. The utility value is the product of the preference weight and the attribute value.

```
Current Date and Time: Fri May 16 18:15:28 2014

Buyers:      5
Sellers:     3

--- Parameters
IMax = 20
InitExpPayoff =    +1.20
Temptation =       +0.60
BothCoop =         +1.00
BothDefect =       +0.60
Sucker =           +0.60
RefusalPayoff =    +0.90
WallflowerPayoff = +0.00
DynamicAlternatePayoffs =  1
ClairvoyantInitialExpectedPayoffs =  0
SellersRankOffersByOrderSize =  0
Verbose =  1
TraceTradesBeginning = 20

--- InitBots
Buyer InitBaselineUtility B01 <- S01
   prefWeight1: -0.0055 attribute1: 39.5 beta_weighted_attribute1: -0.21725
   prefWeight2: -0.045 attribute2: 7 beta_weighted_attribute2: -0.315
   prefWeight3: 0 attribute3: 0 beta_weighted_attribute3: 0
   prefWeight4: 0 attribute4: 0 beta_weighted_attribute4: 0
   prefWeight5: 0 attribute5: 0 beta_weighted_attribute5: 0
  utility: 58.7282
Buyer InitBaselineUtility B01 <- S02
   prefWeight1: -0.0055 attribute1: 45 beta_weighted_attribute1: -0.2475
   prefWeight2: -0.045 attribute2: 7 beta_weighted_attribute2: -0.315
   prefWeight3: 0 attribute3: 0 beta_weighted_attribute3: 0
   prefWeight4: 0 attribute4: 0 beta_weighted_attribute4: 0
   prefWeight5: 0 attribute5: 0 beta_weighted_attribute5: 0
  utility: 56.9783
Buyer InitBaselineUtility B01 <- S03
   prefWeight1: -0.0055 attribute1: 62.25 beta_weighted_attribute1: -
0.342375
   prefWeight2: -0.045 attribute2: 11 beta_weighted_attribute2: -0.495
   prefWeight3: 0 attribute3: 0 beta_weighted_attribute3: 0
   prefWeight4: 0 attribute4: 0 beta_weighted_attribute4: 0
   prefWeight5: 0 attribute5: 0 beta_weighted_attribute5: 0
  utility: 43.2845
```

FIGURE 8: VERBOSE CONSOLE OUTPUTS – INITIALIZATION

In this toy example, with five buyers and three sellers, Buyer 1 has preference weights of -.0055 for Attribute 1, and -.045 for Attribute 2, which refer to cost and shipping time, respectively. The utility to the buyer is calculated as 58.72, which is calculated as

$$\text{Utility} = \exp(-.0055 * 39.5 + -.045 * 7) * 100 = 58.7282$$

As shown below in Figure 9, PMG will then output a list of transactions that take place at each iteration. For each iteration, there will be a list offers and counter-offers, which take place as a results of constraints on the maximum purchases from any single seller, or because a seller has insufficient capacity to fulfill an order. This is followed by a "Trade" section that lists the final set of transactions between buyers and sellers.

```
--- Iteration 20

--- MatchTraders 20

------ SubmitOffers
Buyer B01 Tag CONSTRAINT 0 (S01) reducing offer from 1418750 to 1135000
Buyer B01 asks Seller S01 for 1135000
Buyer B01 asks Seller S02 for 283750
Buyer B02 Tag CONSTRAINT 0 (S03) reducing offer from 102150 to 81720
Buyer B02 asks Seller S03 for 81720
Buyer B02 asks Seller S01 for 20430
Buyer B03 Tag CONSTRAINT 0 (S01) reducing offer from 6810 to 5448
Buyer B03 asks Seller S01 for 5448
Buyer B03 asks Seller S03 for 1362
Buyer B04 Tag CONSTRAINT 0 (S02) reducing offer from 231540 to 185232
Buyer B04 asks Seller S02 for 185232
Buyer B04 asks Seller S01 for 46308
Buyer B05 Tag CONSTRAINT 0 (S02) reducing offer from 17025 to 13620
Buyer B05 asks Seller S02 for 13620
Buyer B05 asks Seller S01 for 3405

------ 1776275 offers

------ SubmitOffers

------ 0 offers

--- Trade 20
Buyer B01 buy from Seller S01 quantity: 1135000
Buyer B04 buy from Seller S01 quantity: 46308
Buyer B05 buy from Seller S01 quantity: 3405
Buyer B02 buy from Seller S01 quantity: 20430
Buyer B03 buy from Seller S01 quantity: 5448
Buyer B01 buy from Seller S02 quantity: 283750
Buyer B04 buy from Seller S02 quantity: 185232
Buyer B05 buy from Seller S02 quantity: 13620
Buyer B02 buy from Seller S03 quantity: 81720
Buyer B03 buy from Seller S03 quantity: 1362
```

FIGURE 9: VERBOSE CONSOLE OUTPUTS – TRANSACTIONS

Once PMG has finished the final iteration, if verbose outputs is selected, PMG will generate a summary of the final set of utilities and trades for each buyer-seller pair. Figure 10 shows summaries for three of the five buyers and their trade interactions with the three sellers in the example. The "them" and "me" rows in the rightmost column show the results of the individual game encounters using cooperate ("c") and defect ("d") notation, making it easy to see where stable trading relationships have arisen and who favored whom, some of which shifted after the first few iterations. For a given iteration, a buyer will offer contracts to as many sellers as needed to fulfill its ordering quota; therefore, some sellers may get no offers from a buyer. Sellers that receive offers despite a "d" defect rating from the buyer are needed to fulfill the ordering needs of the buyer despite offering a lower expected utility than may be had from other sellers.

In this particular example, there were 20 iterations and due to maximum-sourcing fraction constraints of 80%, buyers were forced to choose at least two sellers, resulting in a minimum of 40 transactions per buyer when summed over all iterations. The first summary line for each buyer includes the following data:

- **sumTotPay** is the sum of all payoffs from game results for all 20 iterations for all sellers (e.g. +1864.66 for Buyer 1).
- **numTotPay** is the total number of payoffs (trades executed) for all 20 iterations for all sellers (e.g., 40 for Buyer 1).
- **numW** is not used in PMG and is a is a vestige of the original TNG.
- **fit** refers to a "fitness" score, which is not used in PMG, but is a vestige of the original TNG that used genetic algorithms.

Under each buyer, are records for transaction experiences with each seller, with the following data items:

- **sumPay** is the sum of all payoffs with this same seller for all 20 iterations. These payoffs are calculated exactly as described above for the initial utility calculations, modified by payoff weights (e.g., +1174.56 for Buyer1 and Seller 1).
- **numPay** is the total number of payoffs with this seller (trades executed) for all 20 iterations (e.g., 20p for Buyer 1 and Seller 1).
- **numRef** is not used and is always 0R. It is a is a vestige of the original TNG.
- **expUtility** is the expected utility at the end of the last iteration for transactions with this seller (e.g., 59.29EU for Buyer 1 and Seller 1). This is the expected value of trading with this seller.
- **expPayoff** is the expected payoff (utility) at the end of the last iteration for potential transaction for all sellers *other* than this one (e.g., 35.99ALT for Buyer 1 and Seller 1). This is the expected value of trading with other sellers should this seller not work out.
- **altPayoff** is the total number of alternative payoffs (potential trades, not actual) with other sellers.

Figure 11 shows the same information from the seller's perspective. Here, the one difference is that payoff calculations are based on expected revenues, the product of the unit price and the order size, rather than utilities. Note that transportation and logistics costs are not part of the revenue calculation.

* **Note**: Neither initial nor final utility and trade experience summaries will be output by PMG for very large numbers of buyers and sellers. Only the initial parameters and the information contained in Figure 9 will be output. How this works is unclear, but is possibly due to an integer overflow in the print loop that occurs with a large number of combinations of buyers and sellers.

```
B01:sumTotPay=+1864.66; numTotPay= 40; numW=  0; fit=+93.23
Stored other bot info for this tradebot:
list#;tag;sumPay;numPay;numRef;expUtility;expPayoff;altPayoff;
  1               S01 +1174.56   20P   0R  59.29EU  35.99ALT  20:
                                                                                 them:cccccccccccccccccccc
                                                                                   me:cccccccccccccccccccc

  2               S02 +638.16   18P   0R  37.19EU  46.97ALT  18:
                                                                                 them:cddcdddddddddddddd
                                                                                   me:ccdddddddddddddddd

  3               S03  +51.94    2P   0R  34.63EU  48.33ALT   2:
                                                                                 them:cc
                                                                                   me:dd

B02:sumTotPay=+1941.72; numTotPay= 43; numW=  0; fit=+97.09
Stored other bot info for this tradebot:
list#;tag;sumPay;numPay;numRef;expUtility;expPayoff;altPayoff;
  1               S03 +1083.14   20P   0R  56.57EU  40.91ALT  20:
                                                                                 them:cddddddddddddddddddd
                                                                                   me:cccccccccccccccccccc

  2               S01 +745.30   19P   0R  41.19EU  48.65ALT  19:
                                                                                 them:cdddddddddddddddddd
                                                                                   me:ddddddddddddddddddd

  3               S02 +113.28    4P   0R  40.53EU  49.03ALT   4:
                                                                                 them:cddd
                                                                                   me:dddd

B03:sumTotPay=+2314.65; numTotPay= 40; numW=  0; fit=+115.73
Stored other bot info for this tradebot:
list#;tag;sumPay;numPay;numRef;expUtility;expPayoff;altPayoff;
  1               S01 +1158.48   20P   0R  60.50EU  56.41ALT  20:
                                                                                 them:cddddddddddddddddddd
                                                                                   me:cccccccccccccccccccc

  2               S03 +1156.17   20P   0R  60.38EU  56.47ALT  20:
                                                                                 them:cddddddddddddddddddd
                                                                                   me:cccccccccccccccccccc

  3               S02   +0.00    0P   0R  52.22EU  60.67ALT   0:
                                                                                 them:
                                                                                   me:
```

FIGURE 10: VERBOSE CONSOLE OUTPUTS – TRADING EXPERIENCE SUMMARIES FROM BUYER’S PERSPECTIVE

```
S01:sumTotPay=+117655720.00; numTotPay= 93; numW=  0; fit=+5882786.00
Stored other bot info for this tradebot:
list#;tag;sumPay;numPay;numRef;expUtility;expPayoff;altPayoff;
  1               B01 +113500000.00   20P   0R 5851162.00EU 574333.75ALT  20:
                                                                                  them:cccccccccccccccccccc
                                                                                    me:cccccccccccccccccccc
  2               B04 +2281080.00   16P   0R 685616.50EU 1859380.00ALT  16:
                                                                                  them:dddddddddddddddd
                                                                                    me:ccdddddddddddddd
  3               B02 +1032948.00   19P   0R 520367.47EU 1903193.75ALT  19:
                                                                                  them:ddddddddddddddddddd
                                                                                    me:cdddddddddddddddddd
  4               B05 +503940.00   18P   0R 519912.66EU 1902269.00ALT  18:
                                                                                  them:cccdddcccccccccccd
                                                                                    me:cddddddddddddddddd
  5               B03 +337776.00   20P   0R 462484.66EU 1918128.25ALT  20:
                                                                                  them:cccccccccccccccccccc
                                                                                    me:cddddddddddddddddddd

S02:sumTotPay=+49951248.00; numTotPay= 48; numW=  0; fit=+2497562.50
Stored other bot info for this tradebot:
list#;tag;sumPay;numPay;numRef;expUtility;expPayoff;altPayoff;
  1               B03   +0.00    0P   0R 5222880.00EU 1269566.25ALT   0:
                                                                                  them:
                                                                                    me:
  2               B01 +28602000.00   18P   0R 1780256.88EU 2126777.50ALT  18:
                                                                                  them:ccdddddddddddddddd
                                                                                    me:cddcdddddddddddddd
  3               B04 +20566958.00   20P   0R 1228087.38EU 2265416.50ALT  20:
                                                                                  them:cccccccccccccccccccc
                                                                                    me:cddddddddddddddddddd
  4               B02 +347133.63    4P   0R 1114002.75EU 2296785.50ALT   4:
                                                                                  them:dddd
                                                                                    me:cddd
  5               B05 +435159.00    6P   0R 808291.31EU 2342599.25ALT   6:
                                                                                  them:ccdccc
                                                                                    me:cddddd
```

FIGURE 11: VERBOSE CONSOLE OUTPUTS – TRADING EXPERIENCE SUMMARIES FROM SELLER'S PERSPECTIVE

3.6.9 PMG ALGORITHM

For each iteration, PMG executes the following five basic algorithmic steps:

1. **Buyers order sellers in terms of expected utilities.**
    a. Utilities are a positive function of service attributes, such as delivery time, and transportation and logistics costs, which account for shipment size/frequency and mode. Utility is based on unit costs for a given order and does not depend on the size of the contract, but could be diminished if the seller is unable to fulfill the order (see below). There is no differentiation between sellers based on the quality of the commodity itself, since quality is unknown.
    b. During the first iteration, all buyers have the same expected payoffs (sellers have the same payoff, but buyers' payoff is based on utility). Optionally, one can specify the following condition: ***ClairvoyantInitialExpectedPayoffs*** = 1. If true, then buyers know the size of sellers (capacity), and the sellers know the potential size of orders (demand) from buyers. ClairvoyantInitialExpectedPayoffs may be invoked to reduce the number of iterations required to reach stability and carries with it the assumption of perfect information. A more realistic option might be to allow buyers to obtain advanced information on a limited sample of sellers; however, this has not been implemented, yet.
2. **Buyers offer contracts to the first *n* sellers with the highest expected utilities**, where *n* is the minimum number of sellers required to fill a buyer's demand quota for the commodity.
    a. ***SingleSourceMaxFraction*** is a parameter that may be set to represent risk hedging. For example, it may be assumed that a buyer does not want to purchase more than 80% of a commodity input from a single source, in which case SingleSourceMaxFraction =.8.
    b. If single-source maximum fractions are used, this will require that at least two sellers be selected, unless there is insufficient capacity for available sellers (see below).
3. **Sellers order offered contracts, based on expected revenues** (unit costs * quantity ordered, less transport and logistics costs). This is proportional to order size, except where payoff factors discount revenues (see 3.6.10 below). It is possible to order contracts without regard to payoff factors by setting ***SellersRankOffersByOrderSize*** = 1, which tends to result in quicker convergence as it eliminates the effect of payoff factors for sellers.
4. **Sellers will accept the first n contracts up to their fulfillment capacity.** If a contract could not be completely fulfilled, this is communicated back to the buyer, who is informed that the seller will fulfill a lower quantity. If a seller's capacity is completely used, it will outright reject a contract.
5. **Buyers whose contracts have been diminished or rejected make offers to additional sellers**, drawn from the ordered list formed at the beginning of the iteration. If there does not exist enough additional seller capacity, then the buyer will make offers to the highest rated seller with whom there is an existing contract. In the extreme, a buyer may need to suspend its single-source maximum fractions constraint and offer a larger contract amount to a single seller.

Steps 3-5 are repeated until each buyer is able to fulfill its total demand quota. The program will terminate based on a maximum number of iterations being reached as set in the *.ini file.

While it would be possible to set closure criteria that would terminate the program as the result of a convergent solution having been obtained, early experience with PMG suggests that for very large, realistic markets and problem sizes this may not be computationally tractable, or would lead to excessive run times requiring potentially thousands of iterations. In contrast, for smaller problem sizes, such as experiments with a handful of agents, stable outcomes result in less than one second of run time with 20 iterations or fewer. Therefore, it is recommended that a maximum number of iterations be chosen that provides a solution within a reasonable run time, as determined by the user. Given the real-world opportunities for buying and selling agents to interact and learn about each other and the competition, a limited number of iterations may in fact be more realistic in terms of bounded rationality and imperfect information, while still providing plausible outcomes.

Special behavioral parameters

In addition of the configuration parameters mentioned above in the five algorithmic steps, the following

DynamicAlternatePayoffs, when set to 1 (true), this option invokes a dynamic belief-updating rule, which is generally recommended for consistency with the notion that agent learning takes place. Section 3.6.10 on payoff calculations provides additional insight on this issue. When set to 0 (false), agents will not update their beliefs about the payoff values of trading with agents with whom they have already had a trade encounter. Thus, decisions will be based on initial encounters. In subsequent iterations, new alliances with other agents may form through experimentation, and preferred trading partnerships evolve more slowly.

BuyersIgnoreSoldOutSellers and ***IgnoreSoldOutSellersMinBuyerSellerRatio*** are two "circuit breaker" parameters were added to PMG that work together to allow the program to reduce run times for situations in which the ratio of buyer agents to seller agents is very large. As described above in Step 5, such imbalances cause an excessive amount of offers and counter offers as buyers struggle to find sellers who can meet their purchase requirements within the same iteration. If BuyersIgnoreSoldOutSellers = 1, buyers will not try to trade with sold-out sellers in markets where the ratio of buyers to sellers is above a minimum threshold, as specified by IgnoreSoldOutSellersMinBuyerSellerRatio. In testing PMG, it has been found that setting IgnoreSoldOutSellersMinBuyerSellerRatio = 100 (a ratio of 100:1) tends to work well.

3.6.10 PAYOFF CALCULATIONS

Payoffs are calculated at the end of the iteration once all buyers have found sellers to meet their demand quotas. The payoffs remind the agent what they can expect to receive for future interactions from the same agent and, in the aggregate, what they can expect to receive from the market as a whole. They are stored in agent memory and calculated as successive averages, such as

$$E[U_{it}] = \alpha U_{it} + (1 - \alpha)E[U_{it-1}]$$

where the expected utility, $E[U_{it}]$ of trading with agent *i* resulting from iteration *t* is the α-weighted average of the payoff utility U_{it} obtained at t and the expected utility from the previous iteration, $E[U_{it-1}]$. For simplicity, PMG assumes $\alpha = 0.5$.

- Where no contract offer is made, there is no interaction/experience between buyer and seller, and no game is played. Buyers receive no utility, and sellers receive no revenues. For purposes of future interactions, the expected utility to be received from these agents is unchanged from the previous iteration.

- If a seller rejects an offered contract, buyers receive no utility, and sellers receive no revenues. In addition, there is a rejection payoff penalty applied to the seller by the buyer. For purposes of future interactions, the buyer rates (downgrades) the seller's payoff utility as $U_{it} = V_{it}\pi_{\text{reject}}$ where V_{it} is the un-weighted utility calculation from above, and it is assumed that $0 < \pi_{\text{reject}} < 1$.
- Buyers who offer a contract to a seller whom they have rated as offering utility at least as good as the expected utility value for the market (average across all sellers) are said to "cooperate," meaning they favor this seller.
- Buyers who offer a contract to a seller whom they have rated as offering utility that is lower than the expected utility value for the market average (across all sellers) are said to "defect," meaning they would prefer other sellers.
- Sellers who accept a contract from a buyer whom they have rated as offering revenue at least as good as the expected revenue for the market (average across all sellers) are said to "cooperate," meaning they favor this buyer.
- Sellers who accept a contract from a buyer whom they have rated as offering lower than the expected revenue for the market (average across all sellers) are said to "defect," meaning they would prefer other buyers.
- Payoff values in PMG are weights, which are applied to the nominal utility values that buyers receive from sellers and to the nominal revenue values that sellers receive from buyers. They are used for the purposes of updating agent beliefs about the expected value of future transactions with these same agents. Payoff values are determined by user specified values, which depend on whether both buyer and seller cooperate; both buyer and seller defect; the buyer cooperates and the seller defects; or the buyer defects and the seller cooperates. These values may be arrayed in a 2-by-2 matrix as in Figure 12.

		Seller	
	Outcome	cooperate	defect
Buyer	cooperate	π_a , π_a	π_c , π_b
	defect	π_b , π_c	π_d , π_d

FIGURE 12. GENERAL FORMAT OF A PAYOFF MATRIX

- If both agents cooperate (they both favor trading with each other), the buyer and seller will each assign the value of π_a to the expected return from future trading encounters with this same agent.
    - The buyer rates the seller's payoff utility as $U_{st} = V_{st}\pi_a$, where V_{st} is the un-weighted utility calculation from Step 1.
    - The seller rates the buyer's payoff utility as $U_{bt} = R_{bt}\pi_a$, where R_{bt} is the un-weighted revenue calculation from Step 1.

- If both agents defect (the both believe they would be better off with another trading partner), the buyer and seller will each assign the value of π_d to the expected return from future trading encounters with this same agent.
  - The buyer rates the seller's payoff utility as $U_{st} = V_{st}\pi_d$.
  - The seller rates the buyer's payoff utility as $U_{bt} = R_{bt}\pi_d$.
- If one agent defects and the other cooperates, the defecting agent will assign a value of π_b to the expected utility of future trading encounters with the cooperating agent. In turn, the cooperating agent will assign a value of π_c to the expected utility of trading encounters with the defecting agent. For example, if the buyer were the defector, the buyer would rate the cooperating seller's payoff utility as $U_{st} = V_{st}\pi_b$, and the cooperating seller would rate the buyer's payoff utility as as $U_{bt} = R_{bt}\pi_c$.
- The values of these payoff weights vary based on assumptions of agent information and response dynamics. Different game types results from the values of these weights relative to each other. In a bi-lateral non-cooperative game, such as the classic prisoner's dilemma, the payoffs should be ordered as:

  $\pi_b > \pi_a > \pi_d > \pi_c > 0.$

  Here we have added the restriction that the payoff weights be positive, since PMG applies them as weights on utilities and revenues, rather than at simple face value.

A numerical example in which payoffs have been reformulated as expected values weights in the PMG would be as shown in Figure 13.

		Seller	
	Outcome	cooperate	defect
Buyer	cooperate	1.0 , 1.0	0.5 , 1.5
	defect	1.5 , 0.5	0.7 , 0.7

FIGURE 13: PRISONER'S DILEMMA PAYOFF STRUCTURE

In the parlance of game theory, π_b is classically known as the "***Temptation***" payoff, usually the highest value, while π_c is known as the "***Sucker***" payoff, the lowest value. In the PMG formulation, the payoff values represent adjustments to expected returns for future encounters with the same agent. They do not represent the actual payoffs, which remain the un-weighted utility that the buyer receives from the seller and the un-weighted revenue that the seller receives from the buyer.[5]

Accordingly, the roles of the off-diagonal payoff values are reversed from the usual. π_c is used to weight the expected future utility of a defecting agent by a cooperator, who may or may not know that the defector is defecting. Depending on information assumptions, the cooperating agent may only know that the other agent

[5] Hence, any sort of regularity conditions that have been developed for classic game theory paradigms, such as expected ratios between payoffs in the Prisoner's Dilemma, would be impossible to specify.

offers a higher-than-expected value of utility and thus rates the encounter favorably. π_b is used to weight the expected future utility of a cooperating agent by a defector, who may or may not know that the cooperator is cooperating. Depending on information assumptions, the defecting agent may only know that the other agent offers a lower-than-expected value of utility and thus rates the encounter lower.

Structure "A"		Seller	
	Outcome	Buyer is favored	Buyer is disfavored
Buyer	Seller is favored	1.0 , 1.0	1.0 , 0.7
	Seller is disfavored	0.7 , 1.0	0.7 , 0.7

FIGURE 14: PAYOFF MATRIX WITH ONLY SELF-KNOWLEDGE

Under the assumption that neither agent knows what rating the other agent has assigned to the payoff, only its own rating of the trade, an alternative payoff matrix might look more like the matrix in Figure 14. Here the labels have been changed to more intuitively represent whether the agent views this as a "favored" trading relationship or a "disfavored" relationship, replacing cooperate and defect respectively. The payoff structure in Figure 14 assumes that the rating an agent assigns to the relationship is based only on its own assessment, lacking knowledge of the other agent's views.

A variation in which both agents know what the other agent thinks of the relationship would take on a different format. For example, if the objective of the game were purely to "separate the wheat from the chaff" and there were no bargaining possibilities or price adjustments, then the payoff values might look more like the matrix shown in Figure 15.

Structure "B"		Seller	
	Outcome	Buyer is favored	Buyer is disfavored
Buyer	Seller is favored	1.0 , 1.0	0.7 , 0.7
	Seller is disfavored	0.7 , 0.7	0.7 , 0.7

FIGURE 15: MUTUAL KNOWLEDGE WITHOUT BARGAINING

Agents share their opinions of the trading relationship with each other, but no bargaining or price adjustment occurs. If one agent defects and rates the other agent as disfavored, then both agents know it, and they rate each other with a downgraded payoff for future encounters. Only in the case of mutual cooperation (both favor) are favorable payoff weights received.

Either of the above two payoff schedules, "A" or "B," would be appropriate in instances where the buyers and sellers are assumed to be price-takers; that is, the prices for the goods and services offered by the seller are fixed during the simulation. For forecasting purposes, this may be justified if it were assumed that the computation of transportation and logistics costs and production costs are reasonably accurate, or at least unbiased, and reflect the market in a steady state. The modeling objective would be for the spatial distribution of freight flows to emerge from the simulation under the assumption that the locations of buyers and sellers are fixed, that transportation and logistics costs are accurate, non-transport and logistics costs are accurate, and that inputs to the scenario of total supply and demand are reflected in these costs.

If, however, the objective of the analysis were to discover a different pattern of flows due to a structural change in an influential commodity market or region, or if the analyst simply wants to test how prices might adjust under different scenarios, then a different payoff structure would be needed. The table below is similar to the prisoner's dilemma example shown above, but it takes on a different meaning in PMG due to the use of payoff factors in rating other agents.

Mutual cooperation or defection (favored or disfavored) may be interpreted as before. In the case of mutual cooperation, both agents recognize that they are favored trading partners and rate each other's payoffs at the full value of the utility or revenue. In the case of mutual defection, both agents recognize that they would both prefer to trade with other partners, so they downgrade each other's expected utility/revenues in subsequent iterations.

Structure "C"		Seller	
	Outcome	Buyer is favored	Buyer is disfavored
Buyer	Seller is favored	1.0 , 1.0	0.6 , 1.3
	Seller is disfavored	1.3 , 0.6	0.7 , 0.7

FIGURE 16: MUTUAL KNOWLEDGE WITH BARGAINING

An interesting outcome arises when agent assessments are not mutual. As shown in Figure 16, payoff Structure "C," if one agent feels that this is a favorable relationship and the other does not, we can make the assumption that the cooperator would prefer to trade with this partner in the future and is therefore willing to bargain. For example, if the buyer is the defector and the seller is the cooperator then, in this example, the seller would store the information that in order to trade again with this buyer it might have to lower its price by a factor of 0.67. Simultaneously, the buyer would store the information that it could expect to receive 1.5 times more utility from the seller in future trades. The converse situation in which the buyer is willing to pay more to the seller may also arise in a market in which certain sellers are in high demand and resources are constrained. Played out over multiple buyer-seller combinations and multiple game iterations, the final iteration's set of expected utilities to be received by buyers and the expected revenue to be received by sellers may be interpreted as price adjustments relative to their starting points at Iteration 1. In fact, not interpreting Structure "C" as an instance of bargaining and price adjustment could be irrational, because it would suggest that agents would prefer trading partners that offer them lower utility/revenues.

Role of Payoffs in Feedback and Learning

The payoff values play a role not only in assessing the current trading partner, but also in forming expected values for the utility/revenues that might be obtained in the market overall, since the expected utilities of all agents are aggregated. Thus, when a buyer chooses to downgrade or upgrade a specific seller, that calculation is stored in agent memory and figures into the expected values that are used to decide whether to offer contracts to other sellers, as well. A seller can move up or down in the esteem of a buyer based not only on past encounters with the buyer, but also due to past encounters that the buyer had with other sellers. The complement exists with sellers rating their encounters with buyers and ordering them by expected revenues. Moreover, once a contract has been offered and accepted, how a particular trading partner rates with respect to the average expected utilities/revenues that may be obtained from other potential trading partners (the cooperate vs defect rating) is a function of these past iterations. Thus, the process of adjusting expected utilities/revenues simulates a learning process and facilitates the convergence to stable outcomes.

Special Payoff Configuration Parameters

In addition to the four standard payoff values described above, PMG include three special payoff configuration parameters, which may be set to affect game play.

InitExpPayoff is intended to encourage or discourage experimentation.. To encourage **experimentation**, it should be set to a value greater than the BothCoop payoff value. To discourage experimentation, it should be set to a value lower than the BothCoop value. A lower initial expected payoff setting could represent theorized agent **inertia.** It could also be invoked to simply to reduce the number of iterations needed to establish stable matches between buyer and seller agents.

For example, if the BoothCoop value were set to 1.0 and the InitExpPayoff were set to 1.1 (experimentation), then a buying agent would be more likely to try trading with a seller agent with whom she/he has not interacted previously than a 9%-less expensive seller agent with whom the same buyer had cooperated with in previous rounds. Similarly, if the BoothCoop value were set to 1.0 and the InitExpPayoff were set to 0.9 (inertia), then the buyer agent would not consider new sellers unless they offered at least 10% better value than sellers with whom they have already traded and cooperated.

In subsequent rounds of interactions between the same buyer-seller, the buyer's esteem for the seller will change as a result of these encounters and experiences with other agents. The extent to which the InitExpPayoff will affect subsequent rounds depends on the number of agents participating in the game and the relative competitiveness of their offerings. In a market with a small number of agents, or in which there are one or two dominant suppliers, buyer-seller pairs may quickly settle into a stable pattern of all cooperation or defection after just a few rounds, and the effect of the initial payoff setting will either hasten stable matches (inertia settings) or cause additional experimentation, but not for long. By contrast, in more competitive markets, particularly with many agents, buyers may chose different sellers each round, for as many rounds as there are potentially-viable untested trading partners, unless InitExpPayoff is set to force inertia.

RefusalPayoff is applied only to sellers who refuse an offer from a buyer outright. In general, this only happens in the case of a commodity market in which seller capacity is very constrained. A low refusal payoff would signal to a buyer not to consider in future iterations a seller who spurned it.

WallflowerPayoff is a legacy setting from the original TNG program and is used to penalize sellers who do not participate in any trade alliances. As currently configured, this parameter has no meaning in PMG and does not affect game play.

Variation in Payoff Structures

As currently configured, the meso-scale freight model is set up to generate only a single PMG.ini file, which provides a single set of payoff values for all of the markets. In future enhancements, there could be different sets of payoff values for different commodity markets, differentiated by NAICS codes. In addition, it is expected that the user will want to experiment with different sets of payoff weights to test the effects of information assumptions, classical game theory forms, and theoretical price adjustments, as suggested above.

3.6.11 REFERENCES

Cachon, G. (2003) Supply Chain Coordination with Contracts. In Graves, S. and de Tok, A. Eds. Handbooks in Operations Research and Management Science: Supply Chain Management, Vol. 11: Design, Coordination and Operation. Amsterdam, Elsevier. Chapter 6, pp. 229–329.

Chen, F. (2003) Information Sharing and Supply Chain Coordination. In Graves, S. and de Tok, A. Eds. Handbooks in Operations Research and Management Science: Supply Chain Management, Vol. 11: Design, Coordination and Operation. Amsterdam, Elsevier. Chapter 7, pp. 341–410.

Kranton, R. and Minehart, D. (2001). A Theory of Buyer-Seller Networks. American Economic Review, 91 (3).

Kübler, D, Müller, W. and Normann, H. (2008) Job-Market Signaling and Screening: An Experimental Comparison. Games and Economic Behavior 64: 219–236.

Laffont , J. and Martimort, D. (2002) The Theory of Incentives: The Principal-Agent Model. Princeton University Press.

McFadzean, D., Stewart, D. and Tesfatsion, L. (2001) A Computational Laboratory for Evolutionary Trade Networks. IEEE Transactions on Evolutionary Computation, Vol. 5, No. 3, 546-560.

Rasmusen, E. (2006a) Games and Information: An Introduction to Game Theory, 4th Edition, Wiley: Chapter 11, pp. 341-371. Accessed at http://www.rasmusen.org/GI/download.htm, July 18, 2014.

Rasmusen, E. (2006b) Games and Information: An Introduction to Game Theory, 4th Edition, Wiley: Chapter 7, pp. 195-219. Accessed at http://www.rasmusen.org/GI/download.htm, July 18, 2014.

Stigligtz, J. and Weiss, A. (1990) Sorting Out the Differences Between Signaling and Screening Models. NBER Technical Working Paper No. 93. Accessed at http://www.nber.org/papers/t0093, July 18, 2014.

4.0 MODEL DATABASE

This section of the user's guide provides documentation of three elements of the model:

- Section 4.1 describes the tabular inputs to the model, including descriptions of the purpose of the file, how it is used in the model, its source, and then details of each field in the data.
- Section 0 describes the model's parameters – essentially the inputs to the model that are single values. As with the tabular inputs, the documentation presented below provides a description of the purpose of the parameter, how it is used in the model, and its source.
- Section 4.3 describes the outputs from the model, including summaries and large tabulations that are aggregated from the individual PMG outputs and passed on to the later steps of the model.

The model's inputs, parameters, and outputs are all included within the model's scenario directory structure as shows in Figure 17. Each named scenario contains three items – an inputs folder, and outputs folder, and an R script file called "scenario_variables.R" which holds all of the single value parameters.

```
./scenarios
      /[scenario name]
            /inputs
            /outputs
            scenario_variables.R
      /[scenario name 2]
            …
```

FIGURE 17: FOLDER STRUCTURE FOR SCENARIO INPUTS, OUTPUTS, AND PARAMETERS

4.1 DESCRIPTION OF PRE-PROCESSED DATA INPUTS

The tabular inputs to the model for a particular scenario are included in the ./scenarios/[scenario name]/inputs folder. There are 24 input files used by the model, listed in Table 28. All of the files are comma separated variables (.csv) format files, with dimensions as shown in Table 28 below. The remainder of this section describes each file in more detail, including its use in the model, source, field descriptions, and provides a snapshot of the file's first few rows and fields.

TABLE 28: LIST OF INPUT TABLES

FILENAMES	ROWS	FIELDS	DESCRIPTION
corresp_naics6_n6io_sctg	1175	7	Correspondence between NAICS categories and SCTG commodities categories
corresp_sctg_category	43	5	Correspondence between SCTG categories and commodity group categories used for parameter assumptions
data_2010io	302	387	Detailed 2010 Use table after redefinitions data developed

FILENAMES	ROWS	FIELDS	DESCRIPTION
			using the 2007 benchmark IO account
data_emp_cbp	113409	13	Number of employees and establishments by six digits NAICS industry, FAF zone, and county
data_emp_cbpzone	151	2	Number of employees by zone
data_firm_pref_weights	43	6	Cost and time preference weights and single-source max fraction share by SCTG commodity categories
data_foreign_cons	53486	6	Total exports value, valued on a free alongside ship (FAS) basis, by country and 6-digit NAICS code
data_foreign_prod	28470	6	CIF (cost, insurance, and freight) value of U.S. imports by country, and 6-digit NAICS code
data_mesozone_centroids	269	3	Zone centroid coordinates
data_mesozone_emprankings	132	23	Employment ranking by industry by county
data_mesozone_gcd	223729	7	Great circle distance (GCD) between mesozones
data_mesozone_skims	17424	3	Skims for origins and destinations with available path costs and times
data_modepath_skims	52008	241	Skim costs and times for set of multimodal paths
data_unitcost	43	2	Unit cost (dollar/ton) by two digits SCTG commodity codes
model_distchannel_calibration	33	3	Distribution channel type (number of stops, 0, 1 or 2+ stops) model calibration data: shares by commodity groups
model_distchannel_food	9	5	Distribution channel model variables and coefficients by distributing channel type for food products
model_distchannel_mfg	9	5	Distribution channel model variables and coefficients by distributing channel type for manufactured products
model_numberoftours	20	5	Number of tours model variables and coefficients
model_shipsize_calibration	396	6	Shipment size model calibration data: shares by shipment weight groups and commodity groups for value and tons
model_shipsize_food	9	5	Shipment size model variables and coefficients by shipment size groups for food products
model_shipsize_mfg	9	5	Shipment size model variables and coefficients by shipment size groups for manufactured products
model_stopduration	30	5	Stop duration model variables and coefficients

FILENAMES	ROWS	FIELDS	DESCRIPTION
model_timeofday	13	5	Time of day model variables and coefficients
model_vehicle_tourpattern	26	5	Vehicle and tour pattern model variables and coefficients

4.1.1 CORRESP_NAICS6_N6IO_SCTG.CSV

Description

This file is a correspondence between three classifications and shows the commodities produced by each industry. Industries and commodities are defined at the 6-digit NAICS level using both the systems used by the U.S. Census Bureau and the slightly more aggregated system used by the U.S. Bureau of Economic Analysis. In addition, the table indicates the correspondence between these detailed NAICS 6-digit industry codings and the much more aggregate 2-digit SCTG commodity classification. The table shows the commodities that are produced by each industry. In some cases, industries produce more than one commodity; the final column is a proportion that is used to account for this. In many cases, a single industry is credited with making the entire domestic supply of a commodity (proportion = 1.0), which is expected given the 6-digit level of detail. Where the proportion is less than 1.0, there should be multiple industry entries for the same commodity, such that their proportions sum to 1.0.

Field Definitions

Table 29 describes the fields in the corresp_naics6_n6io_sctg.csv file, while Table 30 shows a snapshot of the file.

TABLE 29: FORMAT OF CORRESP_NAICS6_N6IO_SCTG.CSV

FIELD	UNIT	DESCRIPTION
Industry_NAICS6_Make	-	Six-digit NAICS (BEA) code of the industry
Industry_NAICS6_Make_desc	-	Description of NAICS6_Make industry code
Industry_NAICS6_CBP	-	Six-digit NAICS (Census) commodity code
Industry_NAICS6_CBP_desc	-	Description of the NAICS (Census) commodity code
Commodity_SCTG	-	SCTG two-digit code of the commodity
Commodity_SCTG_desc	-	SCTG code description
Proportion	-	Proportion of the commodity made by the NAICS6_Make industry

TABLE 30: SNAPSHOT OF CORRESP_NAICS6_N6IO_SCTG.CSV

INDUSTRY_NAICS6_MAKE	INDUSTRY_NAICS6_MAKE_DESC	INDUSTRY_NAICS6_CBP	INDUSTRY_NAICS6_CBP_DESC	COMMODITY_SCTG	COMMODITY_SCTG_DESC	PROPORTION
1111A0	Oilseed Farming	111110	Soybean Farming	3	Other ag prods.	1
1111A0	Oilseed Farming	111120	Oilseed (except Soybean) Farming	3	Other ag prods.	1
1111B0	Grain Farming	111130	Dry Pea and Bean Farming	2	Cereal Grains	1
1111B0	Grain Farming	111140	Wheat Farming	2	Cereal Grains	1
1111B0	Grain Farming	111150	Corn Farming	2	Cereal Grains	1
1111B0	Grain Farming	111160	Rice Farming	2	Cereal Grains	1

Usage

This correspondence file is an input to the firm synthesis step to identify what a firm in that industry produces and to convert between the two different systems of NAICS coding (used in employment data and IO data respectively).

Data Sources

The correspondence between commodity NAICS and commodity SCTG codes is based on commodity descriptions. Information about what is produced by particular industries in based on U.S. Bureau of Economic Analysis (2007), BEA IO Make and Use tables (http://www.bea.gov/industry/io_annual.htm) and is derived from the detailed version of the Make Tables/After Redefinitions in the Industry IO Accounts Data.

4.1.2 CORRESP_SCTG_CATEGORY.CSV

Description

This file shows the commodity group categories used for parameter assumptions. Commodities in "Category" are categorized (animals, bulk natural resources, intermediate processed goods, finished goods and others) based on the commodity's physical characteristics. Commodities in "Category2" are categorized (functional, semi-functional, semi-innovative, and innovative) based on the commodity and supply chain characteristics (e.g. demand pattern, time-sensitivity, cost level, etc.).

Field Definitions

Table 31 describes the fields in the corresp_naics6_n6io_sctg.csv file, while Table 32 shows a snapshot of the file.

TABLE 31: FORMAT OF CORRESP_SCTG_CATEGORY.CSV

FIELD	UNIT	DESCRIPTION
Commodity_SCTG	-	Two digits SCTG commodity code
Commodity_SCTG_desc_short	-	Short name of the SCTG commodity
Commodity_SCTG_desc_long	-	SCTG commodity description
Category	-	Commodity category group 1, used in mode and path choice model
Category2	-	Commodity category group 2, used in the assignment of cost and time weights for use in the PMGs

TABLE 32: SNAPSHOT OF CORRESP_SCTG_CATEGORY.CSV

COMMODITY_SCTG	COMMODITY_SCTG_DESC_SHORT	COMMODITY_SCTG_DESC_LONG	CATEGORY	CATEGORY2
1	Live animals/fish	Live animals and live fish	Animals	Functional/Innovative
2	Cereal grains	Cereal grains	Bulk natural resource (BNR)	Functional
3	Other ag prods.	Other agricultural products	Bulk natural resource (BNR)	Functional/Innovative
4	Animal feed	Animal feed and products of animal origin, n.e.c.	Intermediate processed goods (IPG)	Functional
5	Meat/seafood	Meat, fish, seafood, and their preparations	Finished goods (FG)	Functional/Innovative
6	Milled grain prods.	Milled grain products and preparations, bakery products	Finished goods (FG)	Functional

Usage

This table is used to provide a correspondence between the SCTG codes and more aggregate commodity groups used in various model steps. The categories are defined based on physical characteristics or commodity and supply chain characteristics, and specific model parameters are associated with each of these aggregate commodity groups in, for example, the total cost equation used in the model choice model.

Data Sources

The sources of this table include the article by Marshall L. Fisher[6] (1997), "What is the right supply chain for your product?" (http://www.bea.gov/industry/io_annual.htm) which helped with the idea of categorizing product types based on different product characteristics.

4.1.3 DATA_2010IO.CSV

Description

This file contains detailed 2010 Use table after redefinitions data developed using the 2007 benchmark IO accounts. The IO accounts show how industries interact. This table shows the inputs to industry production and the commodities that are consumed by final users. For each production industry, the table reports the value of goods consumed by each buyer industry. The table includes 386 different producing industries showing about 5.7 trillion dollar value of inputs. The highest producing and consuming industries are shown in Table 33 and Table 34, respectively:

TABLE 33: TOP PRODUCING INDUSTRIES

INDUSTRY CODE	INDUSTRY	TOTAL VALUE OF PRODUCTION ($ MILLION)
211000	Oil and gas extraction	504,878
420000	Wholesale trade	500,839
324110	Petroleum refineries	273,172
221100	Electric power generation, transmission, and distribution	172,248
517110	Wired telecommunications carriers	156,553

TABLE 34: TOP CONSUMING INDUSTRIES

INDUSTRY CODE	INDUSTRY	TOTAL VALUE OF CONSUMPTION ($ MILLION)
324110	Petroleum refineries	442,457
S00500	Federal general government (defense)	168,610
531000	Real estate	147,027
420000	Wholesale trade	127,828
517110	Wired telecommunications carriers	104,192

Field Definitions

Table 35 describes the fields in the data_2010io.csv file, while Table 36 shows a snapshot of the file.

[6] *Fisher, Marshall L., "What is the right supply chain for your product?" Harvard business review 75 (1997): 105-117.*

TABLE 35: FORMAT OF DATA_2010IO.CSV

FIELD	UNIT	DESCRIPTION
Industry_NAICS6_MakeUse	-	NAICS (BEA) commodity code of the making industry
X1111A0-S00203	$ Million	Annual values of commodities exchanged between industries, where columns are the using industry

TABLE 36: SNAPSHOT OF DATA_2010IO.CSV

INDUSTRY_NAICS6_MAKEUSE	X1111A0	X1111B0	X111200	X111300	X111400	X111900
1111A0	2584	285	9	0	0	7
1111B0	0	4603	0	0	0	203
111200	0	0	582	8	0	0
111300	0	0	0	147	0	0
111400	0	0	0	0	3085	0
111900	229	755	0	0	0	547

Usage

The model uses this information to identify for each buyer industry the most important commodities that are consumed and their associated supplier industries.

Data Sources

The sources of this table include the U.S. Bureau of Economic Analysis (2007), BEA IO Make and Use tables (http://www.bea.gov/industry/io_annual.htm). It is derived from Make Tables/After Redefinitions in the detailed version from the Industry IO Accounts Data. The 2010 detailed use table (6-digit NAICS codes) was developed by factoring up the 2007 detailed table by growth factors calculated from the 2007 and 2010 summary level (3-digit NAICS codes) tables and it was assumed all 6-digit NAICS industries under a 3-digit NAICS industry grow with the same growth factor from 2007 to 2010.

4.1.4 DATA_EMP_CBP.CSV

Description

This file shows the number of employees and establishments by six digits NAICS industry, FAF zone, and county, although the employment data in this file are derived from the County Business Patterns data are subject to censoring. This number of establishments are divided into eight different employment size groups as follows:

- e1: 1 to 19
- e2: 20 to 99
- e3: 100 to 249
- e4: 250 to 499
- e5: 500 to 999

- e6: 1000 to 2499
- e7: 2500 to 5000
- e8: >5000

Field Definitions

Table 37 describes the fields in the data_emp_cbp.csv file, while Table 38 shows a snapshot of the file.

TABLE 37: FORMAT OF DATA_EMP_CBP.CSV

FIELD	UNIT	DESCRIPTION
Industry_NAICS6_CBP	-	Six digits NAICS (Census) code
FAFZONE	-	FAF zone
CBPZONE	-	County Business Pattern (CBP) zone
employment	Employees	Total number of employees (subject to censoring)
establishment	Establishments	Total number of establishments
e1-e8	Establishments	Total number of establishments in each employment category

TABLE 38: SNAPSHOT OF DATA_EMP_CBP.CSV

INDUSTRY_NAICS6_CBP	FAFZONE	CBPZONE	EMPLOYMENT	ESTABLISHMENT	E1	E2
113310	19	3	3322	497	471	26
115112	19	3	0	8	8	0
212221	19	3	0	1	1	0
212319	19	3	0	6	5	0
212321	19	3	42	33	30	3
221112	19	3	0	12	6	5

Usage

This employment data is used in the firm synthesis step and in conjunction with information from the Make and Use tables to develop a set of synthetic firms characterized with commodities produced and consumption requirements. County-level employment data for the United States outside of CMAP area, in the form of CBP data, are aggregated to a FAF zone resolution, while the county-level data are used with the CMAP area and during the firm synthesis step allocated to the smaller mesozones used in the model.

Data Sources

The sources of this table include the 2010 U.S. Census County Business Pattern data (http://www.census.gov/econ/cbp/). The dataset is an annual series that provides subnational economic data by industry.

4.1.5 DATA_EMP_CBPZONE.CSV

Description

This file shows total employment for each of the zones in the CBP zone system.

Field Definitions

Table 39 describes the fields in the data_emp_cbp.csv file, while Table 40 shows a snapshot of the file.

TABLE 39: FORMAT OF DATA_EMP_CBPZONE.CSV

FIELD	UNIT	DESCRIPTION
COUNTY	-	Zone numbering for CBP zone system (combination of FAF zones and counties)
CBP_EMP	Emplo yee	Total employment

TABLE 40: SNAPSHOT OF DATA_EMP_CBPZONE.CSV

COUNTY	CBP_EMP
1	365984
2	168268
3	523434
4	157840
5	1527454
6	282756

Usage

This employment data is used in the vehicle and tour pattern model to develop the employment by zone data.

Data Sources

The sources of this table include the 2010 U.S. Census County Business Pattern data (http://www.census.gov/econ/cbp/). The dataset is an annual series that provides subnational economic data by industry.

4.1.6 DATA_FIRM_PREF_WEIGHTS.CSV

Description

This file shows the cost and time preference weights and the maximum single-source fraction by SCTG commodity categories. Weights are calculated using the categorization of commodities (functional, semi-functional, semi-innovative, and innovative) which are determined using assumptions on commodity characteristics (e.g. demand pattern, time-sensitivity, and cost level). Commodity cost-weight is determined using FAF dollar/ton values. Commodity time-weight is determined based on commodity supply chain-related characteristics (e.g. demand pattern, lead-time focus, transportation strategy).

Field Definitions

Table 41 describes the fields in the data_firm_pref_weights.csv file, while Table 42 shows a snapshot of the file.

TABLE 41: FORMAT OF DATA_FIRM_PREF_WEIGHTS.CSV

FIELD	UNIT	DESCRIPTION
Commodity_SCTG	-	SCTG commodity code
Commodity_SCTG_desc	-	SCTG commodity description
Commodity_Type	-	Commodity categorization (functional, semi-functional, semi-innovative, or innovative)
CostWeight	%	Cost-weight as a share (calculated based on commodity type)
TimeWeight	%	Time-weight as a share (calculated based on commodity type)
SingleSourceMaxFraction	%	Single-source max fraction as a share (calculated based on commodity type)

TABLE 42: SNAPSHOT OF DATA_FIRM_PREF_WEIGHTS.CSV

COMMODITY_SCTG	COMMODITY_SCTG_DESC	COMMODITY_TYPE	COSTWEIGHT	TIMEWEIGHT	SINGLESOURCEMAXFRACTIO
1	Live animals/fish	Innovative	0.8	0.2	0.8
2	Cereal grains	Semi-Functional	0.6	0.4	0.9
3	Other ag prods.	Semi-Innovative	0.4	0.6	0.9
4	Animal feed	Semi-Functional	0.6	0.4	0.9
5	Meat/seafood	Semi-Innovative	0.4	0.6	0.9
6	Milled grain prods.	Semi-Functional	0.6	0.4	0.9

Usage

This table is used to develop inputs for the PMGs. In the PMG, agents representing producers of an output commodity ("buyers") are instantiated with a quantity of an input commodity to purchase and a set of preference weights that allow them to tradeoff unit costs, service time, and potentially other attributes when considering the utility of a potential trading partner.

Data Sources

The sources of this table include the FHWA's 2007 Freight Analysis Framework data (http://www.ops.fhwa.dot.gov/freight/freight_analysis/faf/). The article by Marshall L. Fisher (1997), "What is the right supply chain for your product?" (http://www.bea.gov/industry/io_annual.htm) is also a source for this input table that helped with the idea of categorizing product types (commodities) based on different product characteristics.

4.1.7 DATA_FOREIGN_CONS.CSV

Description

This file shows the total exports value (domestic and foreign) valued on a free alongside ship (FAS) basis. The FAS reflects transaction price including inland freight, insurance and other charges incurred in placing the merchandise alongside the ship at the port of export. The export value in this input table is reported by 6-digit NAICS code and the country where the goods are to be consumed, further processed, or manufactured as known to the shipper at the time of exportation.

Field Definitions

Table 43 describes the fields in the data_foreign_cons.csv file, while Table 44 shows a snapshot of the file.

TABLE 43: FORMAT OF DATA_FOREIGN_CONS.CSV

FIELD	UNIT	DESCRIPTION
Country	-	Foreign country of export destination names
Commodity_NAICS6	-	6 digit commodity code based on NAICS codes
USExpVal	$	Total exports value
FAFZONE	-	FAF zone of the country of destination
ctrycod	-	Country code the country of destination
CBPZONE	-	CBP zone of the country of destination

TABLE 44: SNAPSHOT OF DATA_FOREIGN_CONS.CSV

COUNTRY	COMMODITY_NAICS6	USEXPVAL	FAFZONE	CTRYCOD	CBPZONE
Afghanistan	111130	1296784	806	1	124
Afghanistan	111140	9894311	806	1	124
Afghanistan	111335	70200	806	1	124
Afghanistan	111421	9250	806	1	124
Afghanistan	111998	27300	806	1	124
Afghanistan	112990	802900	806	1	124

Usage

Since the CBP data does not contain foreign employment data, this table is used to include foreign firms in the model to ensure that international flows between the U.S. and foreign countries can be allocated to consuming firms at the foreign country end. This table is used in firm synthesis to develop and characterize a set of agents located in foreign countries that consume goods.

Data Sources

The sources of this table include the USA Census Trade Online (NAICS-based data) (https://usatrade.census.gov/). Provided by the U.S. Census Bureau's Foreign Trade Division, USA Trade Online provide current and cumulative U.S. export and import data on more than 9,000 export commodities and 17,000 import commodities worldwide. NAICS-based data as part of "Foreign Trade Statistics" data is the most detailed available data on US imports and exports by NAICS codes (6 digits) and by country.

4.1.8 DATA_FOREIGN_PROD.CSV

Description

This file shows the cost, insurance, and freight (CIF) value of U.S. imports by country and 6-digit NAICS code. The CIF value represents the landed value of the merchandise at the first port of arrival in the U.S. It is computed by adding the Customs Value to the aggregate cost of all freight, insurance, and other charges (excluding US import duties) incurred in moving merchandise from alongside the carrier at the port of exportation in the country of export and placing it alongside the carrier at the first port of entry in the U.S.

Field Definitions

Table 45 describes the fields in the data_foreign_prod.csv file, while Table 46 shows a snapshot of the file.

TABLE 45: FORMAT OF DATA_FOREIGN_PROD.CSV

FIELD	UNIT	DESCRIPTION
Country	-	Foreign country of origin names (trading with USA)
Commodity_NAICS6	-	6 digit commodity codes based on NAICS codes
USImpVal	$	Total exports value
FAFZONE	-	FAF zone of the country of origin
Ctrycod	-	Country code the country of origin
CBPZONE	-	Country code the country of origin

TABLE 46: SNAPSHOT OF DATA_FOREIGN_PROD.CSV

COUNTRY	COMMODITY_NAICS6	USIMPVAL	FAFZONE	CTRYCOD	CBPZONE
Afghanistan	111130	3228	806	1	124
Afghanistan	111335	6818	806	1	124
Afghanistan	111339	11042	806	1	124
Afghanistan	111998	1891008	806	1	124
Afghanistan	212399	10650	806	1	124
Afghanistan	311421	13065	806	1	124

Usage

Since the CBP data does not contain foreign employment data, this table is used to include foreign firms in the model to ensure that international flows to the U.S. from foreign countries can be allocated to producing firms at the foreign country end. This table is used in firm synthesis to develop and characterize a set of agents located in foreign countries that produce goods.

Data Sources

The sources of this table include the USA Census Trade Online (NAICS-based data) (https://usatrade.census.gov/). Provided by the U.S. Census Bureau's Foreign Trade Division, USA Trade Online provides current and cumulative U.S. export and import data on more than 9,000 export commodities and 17,000 import commodities worldwide. NAICS-based data as part of "Foreign Trade Statistics" data is the most detailed available data on US imports and exports by NAICS codes (6 digits) and by country.

4.1.9 DATA_MESOZONE_CENTROIDS.CSV

Description

This file contains the centroids of the mesozones.

Field Definitions

Table 47 describes the fields in the data_mesozone_centroids.csv file, while Table 48 shows a snapshot of the file.

TABLE 47: FORMAT OF DATA_MESOZONE_CENTROIDS.CSV

FIELD	UNIT	DESCRIPTION
Stop_zone	-	Mesozone id number
X_coord	-	Centroid x coordinate
Y_coord	-	Centroid y coordinate

TABLE 48: SNAPSHOT OF DATA_MESOZONE_CENTROIDS.CSV

STOP_ZONE	X_COORD	Y_COORD
1	102.966	376.985
2	108.999	376.982
3	115.025	376.968
4	121.071	376.964
5	125.767	375.985
6	101.733	371.197

Usage

This file is used in the stop sequence portion of the truck-touring model.

Data Sources

The data are derived from the mesozone GIS polygon layer used in the meso-scale model.

4.1.10 DATA_MESOZONE_EMPRANKINGS.CSV

Description

This file shows the employment ranking by industry by county. A dataset is prepared from employment data that contains the percentile ranking of each of 21 NAICS categories by mesozonebased on employment numbers in each of those industries. Higher employment numbers implies a higher percentile rank. Within each county, each mesozone is assigned to one of ten percentile-ranking categories. Thresholds of 10 percent are used to determine the categories. Mesozones with the highest employment are classified as Rank 10, zones with the lowest employment Rank 1, and so on. Mesozones are FAF3 zones outside of the region, county-sized zones on the fringes of the region, and township-sized zones in the inner counties.

Field Definitions

Table 49 describes the fields in the data_mesozone_emprankings.csv file, while Table 50 shows a snapshot of the file.

TABLE 49: FORMAT OF DATA_MESOZONE_EMPRANKINGS.CSV

FIELD	UNIT	DESCRIPTION
COUNTY	-	County Federal Information Processing Standards (FIPS) code
MESOZONE	-	Mesozone ID number
rank11 to rank3133	-	Rank of each 2-digit NAICS code industry

TABLE 50: SNAPSHOT OF DATA_MESOZONE_EMPRANKINGS.CSV

COUNTY	MESOZONE	RANK11	RANK21	RANK22	RANK23	RANK3133
17031	1	2	NA	NA	1	1
17031	2	1	NA	NA	3	2
17031	3	3	1	1	5	5
17031	4	1	NA	1	6	3
17031	5	1	NA	NA	1	1
17031	6	NA	1	NA	1	1

Usage

For the purpose of mode choice and simulation of freight traffic, the firms in the CMAP region modeling area are assigned to mesozones within each county. A few county zones correspond to only one TAZ. The other counties correspond to more than one TAZ. TAZs are assigned to firms in these counties based on employment ranking by industry. The model uses percentile rankings to assign larger firms to mesozones with more employment and smaller firms to zones with any employment in their industry classes. The model prevents firms from being assigned to mesozones with no employment in the firm's industry class.

Data Sources

The sources of this table include the 2010 U.S. Census County Business Pattern data (http://www.census.gov/econ/cbp/).

4.1.11 DATA_MESOZONE_GCD.CSV

Description

This file shows the great circle distance (GCD) between the mesozones in the model.

Field Definitions

Table 51 describes the fields in the data_mesozone_gcd.csv file, while Table 52 shows a snapshot of the file.

TABLE 51: FORMAT OF DATA_MESOZONE_GCD.CSV

FIELD	UNIT	DESCRIPTION
Production_zone	-	Production zone ID number
Production_lon	-	Longitude of the Pzone
Production_lat	-	Latitude of the Pzone
Consumption_zone	-	Consumption zone ID number
Consumption_lon	-	Longitude of the Czone
Consumption_lat	-	Latitude of the Czone
GCD	Mile	Great Circle Distance (GCD) between the Pzone and Czone

TABLE 52: SNAPSHOT OF DATA_MESOZONE_GCD.CSV

PRODUCTION_ZONE	PRODUCTION_LON	PRODUCTION_LAT	CONSUMPTION_ZONE	CONSUMPTION_LON	CONSUMPTION_LAT	GCD
1	-1.539033	0.7349715	1	-1.539033	0.7349715	0.000000
2	-1.536992	0.7349669	1	-1.539033	0.7349715	9.647484
3	-1.534953	0.7349577	1	-1.539033	0.7349715	19.283984
4	-1.532908	0.7349486	1	-1.539033	0.7349715	28.951554
5	-1.531321	0.7346943	1	-1.539033	0.7349715	36.494919
6	-1.539453	0.7335137	1	-1.539033	0.7349715	9.497864

Usage

This table is used to calculate distances used as a variable in the distribution channel model.

Data Sources

Distances between zones were estimated using the Haversine formula[7].

4.1.12 DATA_MESOZONE_SKIMS.CSV

Description

This file shows the zone-to-zone travel times between mesozones within the CMAP region.

Field Definitions

Table 53 describes the fields in the data_mesozone_skims.csv file, while Table 54 shows a snapshot of the file.

TABLE 53: FORMAT OF DATA_MESOZONE_SKIMS.CSV

FIELD	UNIT	DESCRIPTION
Origin	-	Origin mesozone ID number
Destination	-	Destination mesozone ID number
Time	Hour	Zone-to-zone travel time

TABLE 54: SNAPSHOT OF DATA_MESOZONE_SKIMS.CSV

ORIGIN	DESTINATION	TIME
1	1	0.0100000
1	2	0.2858333
1	3	0.4055000
1	4	0.5256667
1	5	0.5193333
1	6	0.2626667

Usage

This file is used in the stop sequencing component of the truck-touring model

Data Sources

The skims values in this file are derived from the freight network in CMAP's meso-scale model.

4.1.13 DATA_MODEPATH_SKIMS.CSV

Description

This file contains the skims for origins and destinations with available path costs and times. The skims are based on 54 paths for different modal and route alternatives defined in the meso-scale model. These

[7] *http://en.wikipedia.org/wiki/Haversine_formula*

alternatives included direct modes (such as truck- FTL, truck- LTL, rail, etc.), indirect mode (such as rail-truck, water-truck, etc.), and intermodal facilities (e.g., airports, truck terminals, rail terminals, ports). This table has times and costs associated with all the 54 mode-path choices between all buyer/seller pairs. The development of this file is described in the documentation for the meso-scale model.[8] Table 56 describes the path alternative types and Table 57 describes the terminals referenced in the path alternatives.

Field Definitions

Table 55 describes the fields in the data_modepath_skims.csv file, while Table 58 shows a snapshot of the file.

TABLE 55: FORMAT OF DATA_MODEPATH_SKIMS.CSV

FIELD	UNIT	DESCRIPTION
Origin	-	Origin ID of the zone pair
Destination	-	Destination ID of the zone pair
…	-	(series of fields not used in the model)
Cost 1 : Cost 54	$/ton	Travel Cost per unit weight for each of 54 path alternatives
Time 1 : Time 54	Hours	Travel Time for each of 54 path alternatives

TABLE 56: PATH ALTERNATIVES

PATH NUMBER	PATHTYPE
1	'01 - Water using Port 145'
2	'02 - Water using Port 146'
3	'03 - Carload Direct'
4	'04 - FTL(ExtDray)-Carload remainder'
5	'05 - Carload-FTL with stop at 147'
6	'06 - Carload-FTL with stop at 148'
7	'07 - Carload-FTL with stop at 149'
8	'08 - Carload-FTL with stop at 150'
9	'09 - FTL-Carload-FTL with stop at 147'
10	'10 - FTL-Carload-FTL with stop at 148'
11	'11 - FTL-Carload-FTL with stop at 149'

[8] Cambridge Systematics, Inc. (2011) A Working Demonstration of a Mesoscale Freight Model for the Chicago Region Final Report and User's Guide, prepared for the Chicago Metropolitan Agency for Planning.

PATH NUMBER	PATHTYPE
12	'12 - FTL-Carload-FTL with stop at 150'
13	'13 - IMX Direct'
14	'14 - FTL(ExtDray)-IMX remainder'
15	'15 - IMX-LTL with stop at 147'
16	'16 - IMX-LTL with stop at 148'
17	'17 - IMX-LTL with stop at 149'
18	'18 - IMX-LTL with stop at 150'
19	'19 - IMX-FTL with stop at 147'
20	'20 - IMX-FTL with stop at 148'
21	'21 - IMX-FTL with stop at 149'
22	'22 - IMX-FTL with stop at 150'
23	'23 - FTL-IMX-FTL with stop at 147'
24	'24 - FTL-IMX-FTL with stop at 148'
25	'25 - FTL-IMX-FTL with stop at 149'
26	'26 - FTL-IMX-FTL with stop at 150'
27	'27 - LTL-IMX-LTL with stop at 147'
28	'28 - LTL-IMX-LTL with stop at 148'
29	'29 - LTL-IMX-LTL with stop at 149'
30	'30 - LTL-IMX-LTL with stop at 150'
31	'31 - FTL Direct'
32	'32 - FTL-LTL with stop at 133'
33	'33 - FTL-LTL with stop at 134'
34	'34 - FTL-LTL with stop at 135'
35	'35 - FTL-LTL with stop at 136'
36	'36 - FTL-LTL with stop at 137'
37	'37 - FTL-LTL with stop at 138'
38	'38 - FTL-LTL with stop at 139'
39	'39 - LTL-FTL-LTL with stop at 133'

PATH NUMBER	PATHTYPE
40	'40 - LTL-FTL-LTL with stop at 134'
41	'41 - LTL-FTL-LTL with stop at 135'
42	'42 - LTL-FTL-LTL with stop at 136'
43	'43 - LTL-FTL-LTL with stop at 137'
44	'44 - LTL-FTL-LTL with stop at 138'
45	'45 - LTL-FTL-LTL with stop at 139'
46	'46 - LTL Direct'
47	'47 - Air using Airport 141'
48	'48 - Air using Airport 142'
49	'49 - Air using Airport 143'
50	'50 - Air using Airport 144'
51	'51 - Intl. Water, No Transload, 40 ft. FTL'
52	'52 - Intl. Water, No Transload, 40 ft. LTL'
53	'53 - Intl. Water, Transload, 53 ft. FTL'
54	'54 - Intl. Water, Transload, 53 ft. LTL'

TABLE 57: TERMINALS DESCRIPTIONS

TERMINAL NUMBER	DESCRIPTION
133	'Truck terminal 133'
134	'Truck terminal 134'
135	'Truck terminal 135'
136	'Truck terminal 136'
137	'Truck terminal 137'
138	'Truck terminal 138'
139	'Truck terminal 139'
141	'O''Hare'
142	'Midway'
143	'Gary'

TERMINAL NUMBER	DESCRIPTION
144	'Milwaukee'
145	'Illinois Intl. Port'
146	'Indiana Harbor'
147	'Rockford-area rail yards'
148	'Global III-Rochelle'
149	'Logistics Park - Elwood'
150	'Central Chicago rail yards'

TABLE 58: SNAPSHOT OF DATA_MODEPATH_SKIMS.CSV

ORIGIN	DESTINATION	CA141	CA142	CA143	CA144	TA141
1	1	NA	NA	NA	NA	NA
1	2	NA	NA	NA	NA	NA
1	3	NA	NA	NA	NA	NA
1	4	NA	NA	NA	NA	NA
1	5	NA	NA	NA	NA	NA
1	6	NA	NA	NA	NA	NA

Usage

This table is used in the mode and path choice model to define travel times and cost for each of the 54 path alternatives for travel to, from, and within the CMAP region.

Data Sources

The sources of this table are the networks, cost assumptions, and path-building assumptions as defined in the meso-scale model documentation.

4.1.14 DATA_UNITCOST.CSV

Description

This file shows the unit cost (dollar/ton) by 2-digit SCTG commodity codes. Commodity unit cost is determined using FAF dollar/ton values for all modes by commodity. The total tons carried by all modes all movements are divided to the total value to calculate the unit cost of the commodity.

Field Definitions

TABLE 59 DESCRIBES THE FIELDS IN THE DATA_UNITCOST.CSV FILE, WHILE

Table 60 shows a snapshot of the file.

TABLE 59: FORMAT OF DATA_UNITCOST.CSV

FIELD	UNIT	DESCRIPTION
Commodity_SCTG	-	Two digits SCTG commodity code
UnitCost	$/ton	Commodity unit cost

TABLE 60: SNAPSHOT OF DATA_UNITCOST.CSV

COMMODITY_SCTG	UNITCOST
1	1355.94
2	133.00
3	614.03
4	356.92
5	2816.09
6	1166.85

Usage

This table is used to convert production and consumption values from the BEA IO tables to tonnages produced and consumed.

Data Sources

The sources of this table include the FHWA's 2007 Freight Analysis Framework data (http://www.ops.fhwa.dot.gov/freight/freight_analysis/faf/).

4.1.15 MODEL_DISTCHANNEL_CALIBRATION.CSV

Description

This file shows the "distribution channel type" (number of stops, 0, 1 or 2+ stops) shares by commodity groups. Shipping chain or distribution channel indicates whether the goods went through a consolidation center, a distribution center, and/or a warehouse.

The commodity group codes are aggregations of the 43 SCTG commodity groups as follows:

A: Agricultural Products, B: Chemical/Pharmaceutical products, C: Coal/Mineral/Ores, D: Electronics, E: Prepared Foodstuffs, F: Gravel/Natural Sands/Cements, G: Machinery/Metal Products, H: Mixed Freight/Miscellaneous, I: Motorized and Other Vehicles (incl. parts), J: Wood/Paper/textile/Leather products, K: Other

Field Definitions

Table 61 describes the fields in the model_distchannel_calibration.csv file, while Table 62 shows a snapshot of the file.

TABLE 61: FORMAT OF MODEL_DISTCHANNEL_CALIBRATION.CSV

FIELD	UNIT	DESCRIPTION
Commodity_Category	-	Commodity groups based on the reference paper used
NumberofStops_Choice	-	Shipping chain type
Target_Share	%	Proportion of shipments by distribution channel for each commodity group

TABLE 62: SNAPSHOT OF MODEL_DISTCHANNEL_CALIBRATION.CSV

COMMODITY_CATEGORY	NUMBEROFSTOPS_CHOICE	TARGET_SHARE
A	0	0.50
A	1	0.22
A	2+	0.28
B	0	0.47
B	1	0.36
B	2+	0.17

Usage

This table is used for distribution channel model calibration.

Data Sources

The source of this table is the study[9] done by University of Illinois at Chicago.

4.1.16 MODEL_DISTCHANNEL_FOOD.CSV

Description

This file shows the distribution channel model variables and coefficients by distributing channel type for food products. A multinomial logit (MNL) model was estimated for choice of distribution channel.

[9] *Pourabdollahi, Z., Karimi, B., Mohammadian, A. K., & Kawamura, K. (2014). Shipping Chain Choices in Long Distance Supply Chains: Descriptive Analysis and a Decision Tree Model 2. In Transportation Research Board 93rd Annual Meeting (No. 14-1706).*

Field Definitions

Table 63 describes the fields in the model_distchannel_food.csv file, while Table 64 shows a snapshot of the file.

TABLE 63: FORMAT OF MODEL_DISTCHANNEL_FOOD.CSV

FIELD	UNIT	DESCRIPTION
CHID	-	Distribution channel choice ID
CHDESC	-	Distribution channel choice description
VAR	%	Explanatory variable
TYPE		Type of the explanatory variable
COEFF		Coefficient of the variable

TABLE 64: SNAPSHOT OF MODEL_DISTCHANNEL_FOOD.CSV

CHID	CHDESC	VAR	TYPE	COEFF
1	Direct	asc1	Constant	0.000
1	Direct	emple49	Variable	0.907
2	One	asc2	Constant	-0.932
2	One	mfgind	Variable	1.940
3	Two	asc3	Constant	-3.320
3	Two	trwind	Variable	3.490

Usage
This table is used in the distribution channel step of the model for food commodities.

Data Sources
The sources of this table include the FAME[10] survey data developed by the University of Illinois at Chicago.

4.1.17 MODEL_DISTCHANNEL_MFG.CSV

Description
This file shows the distribution channel model variables and coefficients by distributing channel type for manufactured goods. A multinomial logit (MNL) model was estimated for choice of distribution channel.

---

[10] *Freight Activity Microsimulation Estimator, https://iatbr2009.asu.edu/ocs/custom/resource/W4_R1_Behavioral%20Freight%20Movement%20Modeling.pdf*

Field Definitions

Table 65 describes the fields in the model_distchannel_mfg.csv file, while Table 66 shows a snapshot of the file.

TABLE 65: FORMAT OF MODEL_DISTCHANNEL_MFG.CSV

FIELD	UNIT	DESCRIPTION
CHID	-	Distribution channel choice ID
CHDESC	-	Distribution channel choice description
VAR	%	Explanatory variable
TYPE		Type of the explanatory variable
COEFF		Coefficient of the variable

TABLE 66: SNAPSHOT OF MODEL_DISTCHANNEL_MFG.CSV

CHID	CHDESC	VAR	TYPE	COEFF
1	Direct	asc1	Constant	0.000
2	One	asc2	Constant	-1.960
2	One	empge200	Variable	0.698
2	One	whind	Variable	1.880
3	Two	asc3	Constant	-2.680
3	Two	whind	Variable	1.500

Usage

This table is used in the distribution channel step of the model for all non-food commodities.

Data Sources

The sources of this table is the FAME survey data developed by the University of Illinois at Chicago

4.1.18 MODEL_NUMBEROFTOURS.CSV

Description

This file shows the number of tours model variables and coefficients by number of tours. A multinomial logit (MNL) model was estimated for choice of the number of tours.

Field Definitions

Table 67 describes the fields in the model_numberoftours.csv file, and Table 68 shows a snapshot of the file.

TABLE 67: FORMAT OF MODEL_NUMBEROFTOURS.CSV

FIELD	UNIT	DESCRIPTION
CHID	-	Number of tours model choice ID
CHDESC	-	Number of tours model choice description
VAR	-	Explanatory variable
TYPE	-	Type of the explanatory variable
COEFF	-	Coefficient of the variable

TABLE 68: SNAPSHOT OF MODEL_NUMBEROFTOURS.CSV

CHID	CHDESC	VAR	TYPE	COEFF
1	All stops in one route	asc1	Constant	4.520
1	All stops in one route	Dest_Dist	Variable	2.090
1	All stops in one route	PU_DO_Weight	Variable	-0.375
1	All stops in one route	Dest_Mac	Variable	1.220
1	All stops in one route	Dest_Office	Variable	1.710
1	All stops in one route	Dest_Retail	Variable	1.570

Usage

This table is used in the number of tours component of the truck-touring model.

Data Sources

The number of tours model was estimated using the Texas commercial vehicle survey[11].

4.1.19 MODEL_SHIPSIZE_CALIBRATION.CSV

Description

This file shows the "shipment size" shares by shipment weight groups and commodity groups for value and tons.

Field Definitions

Table 69 describes the fields in the model_shipsize_calibration.csv file, while Table 70 shows a snapshot of the file.

[11] RSG (2012) Tour-based and Supply Chain Freight Forecasting Framework Final Report Framework, developed for the Federal Highway Administration with University of Illinois at Chicago and John Bowman BAA DTFH61-10-R-00013.

TABLE 69: FORMAT OF MODEL_SHIPSIZE_CALIBRATION.CSV

FIELD	UNIT	DESCRIPTION
Commodity_SCTG	-	Two digits SCTG commodity code
Commodity_SCTG_desc	-	Description of SCTG commodity
ShipmentWeight	lbs	Shipment weight category
WeightCategory	-	Shipment weight category ID
ValuePct	%	Percentage of value in each shipment weight category by SCTG commodity
TonsPct	%	Percentage of tons in each shipment weight category by SCTG commodity

TABLE 70: SNAPSHOT OF MODEL_SHIPSIZE_CALIBRATION.CSV

COMMODITY_SCTG	COMMODITY_SCTG_DESC	SHIPMENTWEIGHT	WEIGHTCATEGORY	VALUEPCT	TONSPCT
0	All Commodities (4)	Less than 50 lbs.	1	0.1267	0.0020
0	All Commodities (4)	50 - 99 lbs.	2	0.0333	0.0013
0	All Commodities (4)	100 - 499 lbs.	3	0.0937	0.0077
0	All Commodities (4)	500 - 749 lbs.	4	0.0265	0.0038
0	All Commodities (4)	750 - 999 lbs.	5	0.0195	0.0035
0	All Commodities (4)	1,000 - 9,999 lbs.	6	0.2038	0.0566

Usage

This table is used for shipment size model calibration.

Data Sources

The sources of this table include the Census Bureau and the Bureau of Transportation Statistics (BTS) Commodity Flow Survey (CFS 2007) data. The CFS is the primary source of national and state-level data on domestic freight shipments by American establishments in mining, manufacturing, wholesale, auxiliaries, and selected retail and services trade industries. Data are provided on the types, origins and destinations, values, weights, modes of transport, distance shipped, and ton-miles of commodities shipped.

4.1.20 MODEL_SHIPSIZE_FOOD.CSV

Description

This file shows the shipment size model variables and coefficients by shipment size groups for food products. A multinomial logit (MNL) model was estimated for choice of shipment size.

Field Definitions

Table 71 describes the fields in the model_shipsize_food.csv file, while Table 72 shows a snapshot of the file.

TABLE 71: FORMAT OF MODEL_SHIPSIZE_FOOD.CSV

FIELD	UNIT	DESCRIPTION
CHID	-	Shipment size choice ID
CHDESC	-	Shipment size choice description
VAR	-	Explanatory variable
TYPE	-	Type of the explanatory variable
COEFF	-	Coefficient of the variable

TABLE 72: SNAPSHOT OF MODEL_SHIPSIZE_FOOD.CSV

CHID	CHDESC	VAR	TYPE	COEFF
1	<999	asc1	Constant	0.000
1	<999	SIC1	Variable	5.840
1	<999	SIC2	Variable	0.975
2	999~9999	asc2	Constant	0.546
2	999~9999	DISTCHAN	Variable	-0.788
3	>9999	asc3	Constant	-1.710

Usage

This table is used in the shipment size step of the model for food commodities.

Data Sources

The Texas commercial vehicle survey dataset was used for estimating the discrete choice model.

4.1.21 MODEL_SHIPSIZE_MFG.CSV

Description

This file shows the shipment size model variables and coefficients by shipment size groups for manufactured products. A multinomial logit (MNL) model was estimated for choice of shipment size.

Field Definitions

Table 73 describes the fields in the model_shipsize_mfg.csv file, while Table 74 shows a snapshot of the file.

TABLE 73: FORMAT OF MODEL_SHIPSIZE_MFG.CSV

FIELD	UNIT	DESCRIPTION
CHID	-	Shipment size choice ID
CHDESC	-	Shipment size choice description
VAR	%	Explanatory variable
TYPE		Type of the explanatory variable
COEFF		Coefficient of the variable

TABLE 74: SNAPSHOT OF MODEL_SHIPSIZE_MFG.CSV

CHID	CHDESC	VAR	TYPE	COEFF
1	<999	asc1	Constant	0.000
1	<999	log_dist	Variable	0.150
1	<999	SIC1	Variable	2.270
1	<999	SIC3	Variable	1.980
2	999~9999	asc2	Constant	-0.107
2	999~9999	DISTCHAN	Variable	-0.911

Usage

This table is used in the shipment size step of the mode . The manufactured goods model was applied to all commodities other than food.

Data Sources

The Texas commercial vehicle survey dataset was used for estimating the discrete choice model.

4.1.22 MODEL_STOPDURATION.CSV

Description

This file shows the stop duration model and coefficients. A multinomial logit (MNL) model was estimated for choice of stop duration.

Field Definitions

Table 75 describes the fields in the model_stopduration.csv file, while Table 76 shows a snapshot of the file.

TABLE 75: FORMAT OF MODEL_STOPDURATION.CSV

FIELD	UNIT	DESCRIPTION
CHID	-	Stop duration choice ID
CHDESC	-	Stop duration choice description
VAR	%	Explanatory variable
TYPE		Type of the explanatory variable
COEFF		Coefficient of the variable

TABLE 76: SNAPSHOT OF MODEL_STOPDURATION.CSV

CHID	CHDESC	VAR	TYPE	COEFF
1	<=15 minutes	asc1	Constant	0.0000
2	15 ~ 30 minutes	asc2	Constant	1.1900
2	15 ~ 30 minutes	PU_DO_Weight	Variable	0.0505
2	15 ~ 30 minutes	Nstops	Variable	-0.0463
2	15 ~ 30 minutes	Veh_class_2	Variable	-0.3880
3	30 ~ 45 minutes	asc3	Constant	1.9000

Usage

This table is used in the stop duration component of the truck-touring model.

Data Sources

The Texas commercial vehicle survey dataset was used to estimate the discrete choice model.

4.1.23 MODEL_TIMEOFDAY.CSV

Description

This file shows the time of day model variables and coefficients. A multinomial logit (MNL) model was estimated for choice of time of day.

Field Definitions

Table 77 describes the fields in the model_timeofday.csv file, while Table 78 shows a snapshot of the file.

TABLE 77: FORMAT OF MODEL_TIMEOFDAY.CSV

FIELD	UNIT	DESCRIPTION
CHID	-	Time of day choice ID
CHDESC	-	Time of day choice description
VAR	%	Explanatory variable
TYPE		Type of the explanatory variable
COEFF		Coefficient of the variable

TABLE 78: SNAPSHOT OF MODEL_TIMEOFDAY.CSV

CHID	CHDESC	VAR	TYPE	COEFF
1	<6 AM	asc1	Constant	-5.9700
1	<6 AM	TotPUDOWeight	Variable	0.0454
1	<6 AM	TotStopDur	Variable	0.4290
1	<6 AM	TotTourTime	Variable	0.7060
2	6~8AM	asc2	Constant	-3.1300
2	6~8AM	TotPUDOWeight	Variable	0.0484

Usage

This table is used in the time of day component of the truck-touring model.

Data Sources

The Texas commercial vehicle survey dataset was used to estimate the discrete choice model.

4.1.24 MODEL_VEHICLE_TOURPATTERN.CSV

Description

This file shows the vehicle and tour pattern model variables and coefficients. A multinomial logit (MNL) model is estimated for choice of vehicle and tour pattern.

Field Definitions

Table 79 escribes the fields in the model_vehicle_tourpattern.csv file, while Table 80 shows a snapshot of the file.

TABLE 79: FORMAT OF MODEL_VEHICLE_TOURPATTERN.CSV

FIELD	UNIT	DESCRIPTION
CHID	-	Vehicle and tour pattern choice ID
CHDESC	-	Vehicle and tour pattern choice description
VAR	%	Explanatory variable
TYPE		Type of the explanatory variable
COEFF		Coefficient of the variable

TABLE 80: SNAPSHOT OF MODEL_VEHICLE_TOURPATTERN.CSV

CHID	CHDESC	VAR	TYPE	COEFF
1	Direct, 4 Tires	asc1	Constant	0.000
2	Direct, 6-8 Tires	asc2	Constant	-4.500
2	Direct, 6-8 Tires	DO_Weight	Variable	0.412
2	Direct, 6-8 Tires	PU_Weight	Variable	0.355
3	Direct, Semi/Trailer	asc3	Constant	-4.410
3	Direct, Semi/Trailer	DO_Weight	Variable	0.371

Usage

This table is used in the vehicle and tour pattern component of the truck-touring model

Data Sources

The Texas commercial vehicle survey dataset was used to estimate the discrete choice model.

4.2 MODEL PARAMETERS

The model's parameters for a particular scenario are included in an R script, located at ./scenarios/[scenario name]/scenario_variables.R. There are 63 parameters defined in this file, which are listed in Table 81.

TABLE 81: LIST OF MODEL PARAMETERS DEFINED IN SCENARIO_VARIABLES.R

PARAMETER	COMPONENT	BASEVALUE	DESCRIPTION
outputtable	Model System	FALSE	Should the model save large output tabulations?
outputsummary	Model System	FALSE	Should the model create summary output files?
outputRworkspace	Model System	FALSE	Should the model save R workspaces with

PARAMETER	COMPONENT	BASEVALUE	DESCRIPTION
			all objects currently in memory at the end of each model step?
outputlog	Model System	FALSE	Should the model sink output and messages to a log text file?
outputprofile	Model System	FALSE	Should the model activate profiling?
provalthreshold	Firm Synthesis and Matching	0.8	Threshold for percentage of purchase value for each commodity group met by producers
combinationthreshold	Firm Synthesis and Matching	7e+06	Maximum number of combinations of producers and consumers to enter into a PMG
consprodratiolimit	Firm Synthesis and Matching	1e+06	Limit on ratio of consumers to producers to enter into a PMG
foreignprodcostfactor	Firm Synthesis and Matching	0.9	Producer cost factor for foreign producers (applied to unit costs)
wholesalecostfactor	Firm Synthesis and Matching	1.2	markup factor for wholesalers (applied to unit costs)
B1	Mode Choice	100	Constant unit per order
B4	Mode Choice	2000	Storage costs per unit per year
j	Mode Choice	0.01	Fraction of shipment that is lost or damaged
LT_OrderTime	Mode Choice	10	Expected lead time (time between ordering and replenishment)
sdLT	Mode Choice	1	Standard deviation in lead time
BulkHandFee	Mode Choice	1	Handling charge for bulk goods ($ per ton)
WDCHandFee	Mode Choice	15	Warehouse/DC handling charge ($ per ton)
IMXHandFee	Mode Choice	15	Intermodal lift charge ($ per ton)
TloadHandFee	Mode Choice	10	Transload charge ($ per ton; at international ports only)
AirHandFee	Mode Choice	20	Air cargo handling charge ($ per ton)
WaterRate	Mode Choice	0.005	Line-haul charge, water ($ per ton-mile)
CarloadRate	Mode Choice	0.03	Line-haul charge, carload ($ per ton-mile)

PARAMETER	COMPONENT	BASEVALUE	DESCRIPTION
IMXRate	Mode Choice	0.04	Line-haul charge, intermodal ($ per ton-mile)
AirRate	Mode Choice	3.75	Line-haul charge, air ($ per ton-mile)
LTL53rate	Mode Choice	0.08	Line-haul charge, 53 feet LTL ($ per ton-mile)
FTL53rate	Mode Choice	0.08	Line-haul charge, 53 feet FTL ($ per ton-mile)
LTL40rate	Mode Choice	0.1	Line-haul charge, 40 feet LTL ($ per ton-mile)
FTL40rate	Mode Choice	0.1	Line-haul charge, 40 feet FTL ($ per ton-mile)
ExpressSurcharge	Mode Choice	1.5	Surcharge for direct/express transport (factor)
BulkTime	Mode Choice	72	Handling time at bulk handling facilities (hours)
WDCTime	Mode Choice	12	Handling time at warehouse/DCs (hours)
IMXTime	Mode Choice	24	Handling time at intermodal yards (hours)
TloadTime	Mode Choice	12	Handling time at transload facilities (hours)
AirTime	Mode Choice	12	Handling time at air terminals (hours)
LowDiscRate	Mode Choice	0.01	Low-discount rate
MedDiscRate	Mode Choice	0.05	Medium-discount rate
HighDiscRate	Mode Choice	0.25	High-discount rate
CAP1FTL	Mode Choice	60000	Truckload capacity (pounds)
CAP1Carload	Mode Choice	170000	Carload capacity (pounds)
CAP1Airplane	Mode Choice	50000	Air cargo hold capacity (pounds)
LowMultiplier	Mode Choice	0.5	Safety stock constant for Low category commodities
MediumMultiplier	Mode Choice	1	Safety stock constant for Medium category commodities
HighMultiplier	Mode Choice	2.33	Safety stock constant for High category commodities

PARAMETER	COMPONENT	BASEVALUE	DESCRIPTION
LowVariability	Mode Choice	0.03	Standard deviation in annual flow for Low category commodities
MediumVariability	Mode Choice	0.06	Standard deviation in annual flow for Medium category commodities
HighVariability	Mode Choice	0.09	Standard deviation in annual flow for High category commodities
maxrscriptinstances	Model System	16	Maximum number of separate Rscripts.exe processes that can be run at one time; set subject to limit of computer's available RAM. One is used for monitoring (if monitoring is activated), the remainder for running PMGs.
pmgmonitoring	Model System	TRUE	Should monitoring be activated during running the PMGs to email updates to the model users?
pmgmonfrom	Model System	"colin.smith@rsginc.com"	PMG monitoring email (from email address)
pmgmonto	Model System	"colin.smith@rsginc.com","john.gliebe@rsginc.com"	PMG monitoring email (to email address)
pmgmonsmtp	Model System	"WRJHUBVPW01.i-rsg.com"	PMG monitoring email transmission server information
pmgmoninterval	Model System	3600	PMG monitoring interval (seconds)
pmglogging	Model System	TRUE	Should pmg logging be on?
RandomSeed	PMG Settings (See Section 3.6.6)	41	Random starting seed for the PMGs
IMax	PMG Settings (See Section 3.6.6)	6	Number of iterations for the PMG
Verbose	PMG Settings (See Section 3.6.6)	0	Flag to turn on lots of detail about tradebots in the PMG log
DynamicAlternatePayoffs	PMG Settings (See Sections 3.6.6	1	Recalculate alternate payoffs every iteration based on updated expected

PARAMETER	COMPONENT	BASEVALUE	DESCRIPTION
	and 3.6.9)		payoffs
ClairvoyantInitialExpectedPayoffs	PMG Settings (See Sections 3.6.6 and 3.6.9)	0	Should initial expected tradeoffs know size of other traders?
SellersRankOffersByOrderSize	PMG Settings (See Sections 3.6.6 and 3.6.9)	0	Should sellers accept offers based on order size instead of expected payoff?
InitExpPayoff	PMG Settings (See Sections 3.6.6 and 3.6.10)	1.0	Multiplier on initial expected utility/revenue to encourage experimentation with other traders or to simulate inertia
Temptation	PMG Settings (See Sections 3.6.6 and 3.6.10)	1.5	Payoff utility to the agent who defects while the other cooperates
BothCoop	PMG Settings (See Sections 3.6.6 and 3.6.10)	1.0	Payoff utility in which both agents cooperate
BothDefect	PMG Settings (See Sections 3.6.6 and 3.6.10)	0.8	Payoff utility in which both agents defect
Sucker	PMG Settings (See Sections 3.6.6 and 3.6.10)	0.67	Payoff utility to the agent who cooperates while the other defects
RefusalPayoff	PMG Settings (See Sections 3.6.6 and 3.6.10)	0.5	Amount to downgrade expected payoff of seller who outright refuses a trade offer by buyer
WallflowerPayoff	PMG Settings (See Sections 3.6.6 and 3.6.10)	0	Negative payoff to sellers for not participating (not used in PMG)
BuyersIgnoreSoldOutSellers	PMG Settings (See Section 3.6.6)	1	Buyers do not try to trade with sold-out sellers
IgnoreSoldOutSellersMinBuyerSellerRatio	PMG Settings (See Section 3.6.6)	100	Ratio at which buyers do not try to trade with sold-out sellers
RawFastParser	PMG Settings (See Section 3.6.6)	1	Faster reading of input files but does less checks (ok for use with R)
wgtmax_2axl	Vehicle Tour Choice Pattern	35000	Maximum load weight of light duty truck in pounds

PARAMETER	COMPONENT	BASEVALUE	DESCRIPTION
wgtmax_3axl	Vehicle Tour Choice Pattern	65000	Maximum load weight of medium duty truck in pounds
wgtmax_semi	Vehicle Tour Choice Pattern	100000	Maximum load weight of heavy duty truck in pounds
annualfactor	Daily Sample	310	Annualization factor to select a daily sample from the annual shipments

4.3 DESCRIPTION OF MODEL OUTPUTS

The tabular outputs from the model for a particular scenario are included in the ./scenarios/[scenario name]/outputs folder. The output files are either .csv format files, or R binary format files (.Rdata) that contain compressed tables. The types of output are described in Table 82. The remainder of this section describes the output files in more detail, including field descriptions, and provides a snapshot of the file's first few rows and fields.

TABLE 82: TYPES OF MODEL OUTPUT

FILENAME	TYPE OF OUTPUT	DESCRIPTION
XXXXXX.Rdata	R Binary file containing tables	A file for each NAICS market (where XXXXXX is the 6-digit NAICS code) containing the tables of producers and consumers for each market, which is output from firm synthesis. The inputs to the individual PMGs are built from these files. The pairs table of final trades for all of the PMGs covering this market is added to this Rdata file after the PMGs are run
XXXXXX.txt	Log text file	Log file showing start and end times for each PMG run for this NAICS market (where XXXXXX is the 6-digit NAICS code)
XXXXXX_PMG_Log.txt	Log text file	More detailed log file showing the print messages produced during the creation of PMG inputs, calls to run the PMG model, and processing of outputs for this NAICS market (where XXXXXX is the 6-digit NAICS code)
XXXXXX_gY.csv	PMG output (CSV table)	Output csv file from a particular PMG run for a NAICS market (where XXXXXX is the 6-digit NAICS code) and group, where Y is the group number
Firm_syn.txt	Summary text file	A summary text file showing tabulations from the firm synthesis model
Naics_set.Rdata	R Binary file containing	A summary of the numbers of producers, consumers, matches, and numbers of PMG groups for each NAICS market

FILENAME	TYPE OF OUTPUT	DESCRIPTION
	tables	
Pairs.Rdata	R Binary file containing a table	A combined pairs table of final trades for all of the markets, appending together the pairs tables from each NAICS market
Pairs.csv	CSV Table	A combined pairs table of final trades for all of the markets, appending together the pairs tables from each NAICS market
PMG_Workspace.Rdata	R Binary file containing workspace objects	A workspace containing function, variables and other data from the early portion of the R model that is used in each of the separate R script processes to support development of the inputs to the PMGs
stopdur_durationbytourtype.csv	CSV Table	Frequency of stop duration categories by tour types
stopseq_numstopspertour.csv	CSV Table	Frequency of each stop count category
stopseq_tourcatbyshipsize.csv	CSV Table	Frequency of tour category by shipment size
stopseq_stopcluster_sample.png	PNG File	Sample chart of stop clustering for each tour type category
vehtourpat_allcommodities.csv	CSV Table	Frequency of each alternative (combination of tour patterns and vehicle types) for all commodities
vehtourpat_food.csv	CSV Table	Frequency of each alternative (combination of tour patterns and vehicle types) for food commodity group
vehtourpat_mfg.csv	CSV Table	Frequency of each alternative (combination of tour patterns and vehicle types) for manufacturing commodity group
tod_todbytourtype.csv	CSV Table	Frequency of time of day categories by tour types
trip_table.csv	CSV Table	Number of trips by origin mesozone, destination mesozone, vehicle type, and time of day
Warehouses.Rdata	R Binary files containing a table	A list of warehouse locations summarized from the firm synthesis for use in the warehouse allocation model

4.3.1 XXXXXX.RDATA

Description

This set of output files, initially from the firm synthesis model component and ultimately from the PMG control model component, contains the tables of producers and consumers for each NAICS market (where XXXXXX is the 6-digit NAICS code). The inputs to the individual PMGs are built from these files. During the running of the PMG games during the PMG control model component, the producers and consumers tables are updated to add the sample groups that each producer and consumer participate in, and finally the pairs table containing the details of the final trades for the NAICS market is added to the file. These output files are used in the PMG outputs step, where the individual pairs tables from each NAICS markets are combined together into the single large pairs table.

Table and Field Definitions

These Rdata files contain three separate data tables:

- prodc – table of producers who produce commodity XXXXXX
- consc – table of consumers who purchase commodity XXXXXX
- pairs – table of final trades from the PMGs for commodity XXXXXX

Table 83 describes the fields in the prodc table, while Table 84 shows a snapshot of the prodc table. Table 85 and Table 86 show the layout and snapshot of the consc table, while Table 87 and Table 88 show the layout and snapshot of the pairs table.

TABLE 83: FORMAT OF XXXXXX.RDATA, PRODC TABLE

FIELD	UNIT	DESCRIPTION
Commodity_SCTG	-	SCTG commodity code
NAICS	-	NAICS code for synthetic firm
SIZE	-	Number of employees in firm
SELLERID	-	Seller ID number in population of synthetic firms
ZONE	-	Zone ID number in model database (synthetic firm's geographic location)
NonTransportUnitCost	$/pound	Production cost in dollars per pound for this firm and commodity code (does not include transport and logistics costs)
OutputCapacityTons	pounds	Synthetic firm's annual total production/sales capacity in pounds for this commodity code
OutputCommodity	-	NAICS code for commodity produced and sold by this firm (seller)
numgroup	-	Number of sample groups for this NAICS market
group	-	Sample group the producer is assigned to

TABLE 84: SNAPSHOT OF XXXXXX.RDATA, PRODC TABLE

	Commodity_SCTG	NAICS	Size	SellerID	Zone	NonTransportUnitCost	OutputCapacityTons	OutputCommodity	numgroups	group
1:	25	113000	0	7373499	275	0.0192915	524376.0	113000	7	1
2:	25	113000	0	7373503	290	0.0192915	586216.7	113000	7	2
3:	25	113000	0	7373521	335	0.0192915	2595443.6	113000	7	3
4:	25	113000	0	7373523	340	0.0192915	3380556.2	113000	7	4
5:	25	113000	0	7373525	344	0.0192915	3967757.8	113000	7	5
6:	25	113000	0	7373541	373	0.0192915	3678822.3	113000	7	6

TABLE 85: FORMAT OF XXXXXX.RDATA, CONSC TABLE

FIELD	UNIT	DESCRIPTION
Commodity_SCTG	-	SCTG commodity code of commodity required
InputCommodity	-	NAICS code for commodity to be procured (consumed)
NAICS	-	NAICS code for synthetic firm
Zone	-	Zone ID number in model database (synthetic firm's geographic location)
Buyer.SCTG	-	SCTG commodity code produced by the buyer
BuyerID	-	Buyer ID number in population of synthetic firms
Size	-	Number of employees in firm
ConVal	$	Value of commodity needed to purchase ($)
PurchaseAmountTons	pounds	Total annual amount needed to purchase (pounds)
PrefWeight1_UnitCost	-	Proportional utility weight on Attribute1_UnitCost in Costs file (producer cost plus transport and logistics costs, normalized by quantity)
PrefWeight2_ShipTime	-	Proportional utility weight on shipping time (days in the Cost file)
SingleSourceMaxFraction	-	Maximum proportion of purchase quantity from a single seller agent. Valid range: 0 < x <= 1.0. Default is 1.0.
OutputCommodity	-	NAICS code for commodity to being produced by this firm (usually but not always same as firm's NAICS code)
numgroups	-	Number of sample groups for this NAICS market
group	-	Sample group the producer is assigned to

TABLE 86: SNAPSHOT OF XXXXXX.RDATA, CONSC TABLE

	Commodity_SCTG	InputCommodity	NAICS	Zone	Buyer.SCTG	BuyerID	Size	ConVal	PurchaseAmountTons	PrefWeight1_UnitCost	PrefWeight2_ShipTime
1:	25	113000	113000	274	0	7387425	0	8520.8880	397522.19	0.8	0.2
2:	25	113000	311930	274	0	7387426	0	247.1030	11528.01	0.8	0.2
3:	25	113000	321100	274	0	7387427	0	18366.1370	856829.34	0.8	0.2
4:	25	113000	321200	274	0	7387428	0	4911.5187	229135.47	0.8	0.2
5:	25	113000	321910	274	0	7387429	0	2987.4472	139372.39	0.8	0.2
6:	25	113000	3219A0	274	0	7387430	0	788.5084	36786.02	0.8	0.2

	SingleSourceMaxFraction	OutputCommodity	numgroups	group
1:	1	113000	7	1
2:	1	311930	7	2
3:	1	321100	7	3
4:	1	321200	7	4
5:	1	321910	7	5
6:	1	3219A0	7	6

TABLE 87: FORMAT OF XXXXXX.RDATA, PAIRS TABLE

FIELD	UNIT	DESCRIPTION
BuyerID	-	A sequential ID numbers for the buying firm (the firms IDs range from 1 to N where N is the total number of firms synthesized by the model)
SellerID	-	A sequential ID numbers for the selling firm (the firms IDs range from 1 to N where N is the total number of firms synthesized by the model)
Production_zone	-	Production zone, the zone where the selling firm is located, using the Mesozone system
Consumption_zone	-	Consumption zone, the zone where the buying firm is located, using the Mesozone system
NAICS	-	NAICS code of the selling firm and the commodity being traded, using the 6-digit US BEA system as used for the IO tables
Seller.NAICS	-	NAICS code of the seller firm, using the 6-digit US BEA system as used for the IO tables
Commodity_SCTG	-	Standard Classification for Transported Goods, for the commodity being traded
Seller.Size	Employee	Size of the selling firm in number of employees (which represent the mid-points of firm size ranges from the county business patterns data)
OutputCapacityTons	Pounds	Annual Output Capacity in POUNDS of the selling firm (Note that we are working in POUNDS and not tons—changed to avoid rounding)
Buyer.NAICS	-	NAICS code of the buying firm, using the 6-digit US BEA system as used for the IO tables
Buyer.SCTG	-	Standard Classification for Transported Goods, for the commodity produced by the buying firm; zero for firm that does not produce any transported goods
Buyer.Size	employee	Size of the buying firm in number of employees (which represent the mid-points of firm size ranges from the county business patterns data)
ConVal	$	Consumption value: Annual value in $ of the commodity being traded required by the buying firm to satisfy their production requirements
PurchaseAmountTons	Pounds	Annual purchase amount of the commodity being traded in POUNDS required by the buying firm to satisfy their production requirements
Distance	Miles	Great circle distance (GCD) in miles between the selling and buying firms
distchannel	-	Distribution channel assigned to this pair of firms by the distribution channel model (1=direct, 2=1 stop/transfer, 3=2 stops/transfers, 4=3

FIELD	UNIT	DESCRIPTION
		stops/transfers)
ship_size	-	Shipment size category of the individual shipment between this seller and buyer (1=<1000 lbs, 2=1000 to 9999 lbs, 3=10000 or more pounds)
weight	Pounds	Shipment size of the individual shipments between this seller and buyer in pounds
lssbd	-	Flag for large seller, small buyer, and distance of more than 300 miles between them
MinGmnql	$	Logistics cost for minimum path
MinPath	-	Minimum path assigned by the mode choice model
Attribute2_ShipTime	Days	Shipment travel time from seller to buyer in days
Attribute1_UnitCost	$/Pound	Unit cost of shipment transportation
Quantity.Traded	Pounds	The quantity in POUNDS traded between this seller and buyer in this iteration of the PMGs (note the table has been filtered down to just the records with a trade a in the final iteration so the value in this field is the same as the Last.Iteration.Quantity
Number.of.Trades	-	The number of trades between this seller and buyer
Last.Iteration.Quantity	Pounds	The quantity in POUNDS traded between this seller and buyer in the final iteration of the PMGs

TABLE 88: SNAPSHOT OF XXXXXX.RDATA, PAIRS TABLE

```
   BuyerID SellerID Production_zone Consumption_zone  NAICS Seller.NAICS Commodity_SCTG Seller.Size OutputCapacityTons Buyer.NAICS Buyer.SCTG Buyer.Size
1:       2  7373544             385              151 113000       113000             25         794         6736398051      113000         25         10
2:       9  6381951             205              153 113000       113000             25          60          509275347      113000         25         10
3:      16  6381951             205              153 113000       113000             25          60          509275347      113000         25         10
4:      23  6382147             272              153 113000       113000             25          60          509275347      113000         25         10
5:      23  7373544             385              153 113000       113000             25         794         6736398051      113000         25         10
6:      30     7472             265              152 113000       113000             25          10           84879225      113000         25         10
     ConVal PurchaseAmountTons Distance distchannel ship_size weight lssbd   MinGmnql MinPath Attribute2_ShipTime Attribute1_UnitCost Quantity.Traded
1: 271053.2           12645356 8455.902           1         3  75000     0 9570046654      31           4.3688828            756.8032    2.017714e-316
2: 271053.2           12645356 1215.632           2         3  75000     0 1375803396      31           0.6280767            108.7991    1.249527e-316
3: 271053.2           12645356 1215.632           2         3  75000     0 1375803396      31           0.6280767            108.7991    6.247636e-317
4: 271053.2           12645356 1340.021           3         3  75000     0 1516580796      31           0.6923439            119.9318    5.954400e-317
5: 271053.2           12645356 8448.815           1         3  75000     0 9562025314      31           4.3652209            756.1689    1.150580e-316
6: 271053.2           12645356 1249.435           1         3  75000     0 1414060127      31           0.6455416            111.8245    1.538012e-317
   Number.of.Trades Last.Iteration.Quantity
1:                4           6.247636e-317
2:                2           6.247636e-317
3:                1           6.247636e-317
4:                1           5.954400e-317
5:                3           2.932359e-318
6:                1           1.538012e-317
```

4.3.2 XXXXXX.TXT (6-DIGIT NAICS CODE)

Description

These output files are log file showing start and end times for each sample group PMG run for the NAICS market (where XXXXXX is the 6-digit NAICS code). The PMG control model component produces these outputs.

Field Definitions

Table 89 shows a snapshot of one of the output files.

TABLE 89: SNAPSHOT OF XXXXXX.TXT

```
Starting: 113000 Current time 2015-04-08 12:43:15
Starting: 113000 Group 1 Current time 2015-04-08 12:45:08
Completed: 113000 Group 1 Current time 2015-04-08 12:47:56
Starting: 113000 Group 2 Current time 2015-04-08 12:49:57
Completed: 113000 Group 2 Current time 2015-04-08 13:12:05
Starting: 113000 Group 3 Current time 2015-04-08 13:14:02
Completed: 113000 Group 3 Current time 2015-04-08 13:33:53
Starting: 113000 Group 4 Current time 2015-04-08 13:35:50
Completed: 113000 Group 4 Current time 2015-04-08 13:39:05
Starting: 113000 Group 5 Current time 2015-04-08 13:41:01
Completed: 113000 Group 5 Current time 2015-04-08 13:46:51
Starting: 113000 Group 6 Current time 2015-04-08 13:48:48
Completed: 113000 Group 6 Current time 2015-04-08 14:29:04
Starting: 113000 Group 7 Current time 2015-04-08 14:31:01
Completed: 113000 Group 7 Current time 2015-04-08 14:41:48
Completed Processing Outputs: 113000 Current time 2015-04-08 14:41:58
```

4.3.3 XXXXXX_PMG_LOG.TXT (6-DIGIT NAICS CODE)

Description

This set of output files are more detailed log files showing the print messages produced during the creation of PMG inputs, calls to run the PMG model, and processing of outputs for this NAICS market (where XXXXXX is the 6-digit NAICS code). The PMG control model component produces these outputs.

Field Definitions

Table 90 shows a snapshot of one of the output files.

TABLE 90: SNAPSHOT OF XXXXXX_PMG_LOG.TXT

```
[1] "2 to 15 -4481.34430780924"
[1] "2 to 15 21193.6976798327"
[1] "2 to 15 20593.698076971"
[1] "2 to 15 20257.6238901237"
[1] "2 to 15 15517.6601008401"
[1] "2 to 15 15053.1560958248"
[1] "2 to 15 14453.1564929631"
[1] "2 to 15 11610.416713735"
[1] "2 to 15 -16829.3660219668"
warning messages:
1: In `[.data.table`(consc, , `:=`(group, 1:groups)) :
  Supplied 15 items to be assigned to 565390 items of column 'group' (recycled leaving remainder of 10 items).
2: In `[.data.table`(prodc, , `:=`(group, 1:groups)) :
  Supplied 15 items to be assigned to 2521 items of column 'group' (recycled leaving remainder of 1 items).
[1] "Making Inputs: 114000 Group 1 Current time 2015-04-08 12:44:26"
[1] "Writing buy and sell files for  114000 group 1"
[1] "Applying distribution, shipment, and mode-path models to 114000 group 1"
[1] "Organizing data for distribution channel model"
[1] "Applying distribution channel model"
```

4.3.4 XXXXXX_GY.CSV (6-DIGIT NAICS CODE)

Description

This set of output files are the output tables from a particular PMG run for a NAICS market (where XXXXXX is the 6-digit NAICS code) and group, where Y is the group number. The PMG control model component produces these outputs.

Field Definitions

Table 91 describes the fields in the XXXXXX_gY.csv file, while Table 92 shows a snapshot of one of the output files.

TABLE 91: FORMAT OF XXXXXX_GY.CSV

FIELD	UNIT	DESCRIPTION
BuyerId	-	Buyer ID number in population of synthetic firms
SellerId	-	Seller ID number in population of synthetic firms
Quantity.Traded	pounds	The quantity in POUNDS traded between the seller and buyer
Number.of.Trades	-	The number of trades between the seller and buyer
Last.Iteration.Quantity	pounds	The quantity in POUNDS traded between the seller and buyer

TABLE 92: SNAPSHOT OF XXXXXX_GY.CSV

	A	B	C	D	E
1	BuyerId	SellerId	Quantity.Traded	Number.of.Trades	Last.Iteration.Quantity
2	7387425	6381727	397522	1	0
3	7387425	6381860	397522	1	397522
4	7387425	6381867	397522	1	0
5	7387425	6381979	397522	1	0
6	7387425	6382070	397522	1	0

4.3.5 XXXXXX_GY.TXT (6-DIGIT NAICS CODE)

Description

This set of output files are the output tables from a particular PMG run for a NAICS market (where XXXXXX is the 6-digit NAICS code) and group, where Y is the group number. The PMG control model component produces these outputs.

Field Definitions

Table 93 describes the fields in the XXXXXX_gY.txt file, while Table 94 shows a snapshot of one of the output files.

TABLE 93: FORMAT OF XXXXXX_GY.TXT

FIELD	UNIT	DESCRIPTION
BuyerId	-	Buyer ID number in population of synthetic firms
SellerId	-	Seller ID number in population of synthetic firms

FIELD	UNIT	DESCRIPTION
Quantity.Traded	pounds	The quantity in POUNDS traded between the seller and buyer
Number.of.Trades	-	The number of trades between the seller and buyer
Last.Iteration.Quantity	pounds	The quantity in POUNDS traded between the seller and buyer

TABLE 94: SNAPSHOT OF XXXXXX_GY.TXT

```
Current Date and Time: Wed Apr 08 12:45:09 2015

--- Parameters
ini file =         .\PMG\pmg.ini
prefix =           113000_g1
data directory =   E:\Projects\Clients\CMAP\chicago-agent\scenarios\base\outputs
output directory = E:\Projects\Clients\CMAP\chicago-agent\scenarios\base\outputs
IMax = 6
DynamicAlternatePayoffs =  1
ClairvoyantInitialExpectedPayoffs =  1
SellersRankOffersByOrderSize =  1
BuyersIgnoreSoldOutSellers = 1
IgnoreSoldOutSellersMinBuyerSellerRatio = +100.00
RawFastParser =  1
Verbose =  0
TraceTradesBeginning =  6
InitExpPayoff =    +0.90
```

4.3.6 FIRM_SYN.TXT

Description

This output file is a summary text file showing tabulations from the firm synthesis model.

Field Definitions

The text file contains several individual summary values and summary tables. These are:

- firms – total number of establishments synthesized in the firm synthesis mode
- employment – total number of employees in the synthesized establishments
- firmsempbysctg - establishments and employees by SCTG commodity code
- total_value – total value of production contained in the IO table
- Industry_NAICS_Make – total number of NAICS making industries
- Industry_NAICS_Use – total number of NAICS using industries
- producers – total number of producers
- producers_emp – total number of employees of all producers
- producers_cap – total output capacity of all producers
- producersempbysctg – total number of producers, producers employment, and output capacity by SCTG commodity
- producersdomfor – number of domestic and foreign producers

- consumers – total number of consumers
- consumption_pairs – number of rows in the consumers table, indicating the total number of input requirements across all consumers
- threshold – provalthreshold value used for this scenario
- consumer_inputs – total purchase requirements for all consumers
- consumersbysctg – total consumers by SCTG commodity
- consumers_domfor – domestic and foreign number of consumers
- matches – total number of producer/consumer matches by SCTG commodity
- producersempbynaics – total employees and producers by NAICS code
- consumersbynaics – total employees and consumers by NAICS code
- matches_naics – total number of producer/consumer matches by NAICS code
- io_sum_make_naics – total production value by NAICS code from the IO table
- io_sum_make_sctg – total production value by NAICS code from the IO table

Table 95 shows a snapshot of one of the output files.

TABLE 95: SNAPSHOT OF FIRM_SYN.TXT

```
$firms
[1] 7367670

$employment
[1] 167147765

$firmsempbysctg
    Commodity_SCTG              SCTG_Name Establishments Employment
 1:              0                     NA        6608591  143519995
 2:              1      Live animals/fish           1480      25880
 3:              2          Cereal grains           1529      26145
 4:              3         Other ag prods.          2441      81545
 5:              4            Animal feed           3601     112725
 6:              5          Meat/seafood           14563     902775
 7:              6     Milled grain prods.         18609     645145
 8:              7        Other foodstuffs         16903     908855
 9:              8     Alcoholic beverages          7443     310130
10:              9          Tobacco prods.          8343     277675
```

4.3.7 NAICS_SET.RDATA

Description

This output file has a summary of the numbers of producers, consumers, matches, and numbers of PMG groups for each NAICS market. The firm synthesis model component produces this output.

Field Definitions

Table 96 describes the fields in the Naics_set.Rdata file, while Table 97 shows a snapshot of one of the output files.

TABLE 96: FORMAT OF NAICS_SET.RDATA

FIELD	UNIT	DESCRIPTION
NAICS	-	NAICS Industry Code
Producers	-	Total producers
Consumers	-	Total consumers
Possible_Matches	-	Total combinations of producers and consumers
ConsProd_Ratio	-	Ratio of consumers to producers
Split_Prod	-	Flag indicating whether to split producers into separate groups or not
nProducers	-	Number of producers in each group
nConsumers	-	Number of consumers in each group
nMatches	-	Number of combinations of producers and consumers in each group
rev_CPRatio	-	Revised ratio of consumers to producers Following allocation to groups
groups	-	Number of groups

TABLE 97: SNAPSHOT OF NAICS_SET.RDATA

	NAICS	Producers	Consumers	Possible_Matches	ConsProd_Ratio	Split_Prod	nProducers	nConsumers	nMatches	rev_CPRatio	groups
1	113000	8930	37816	337696880	4.234714	TRUE	1276	5403	6894228	4.234326	7
2	114000	2521	565390	1425348190	224.272114	TRUE	169	37693	6370117	223.035503	15
3	211000	9699	29445	285587055	3.035880	TRUE	1386	4207	5830902	3.035354	7
4	212100	10425	336199	3504874575	32.249305	TRUE	454	14618	6636572	32.198238	23
5	212230	66	695766	45920556	10541.909091	TRUE	22	231922	5102284	10541.909091	3
6	2122A0	2966	25130	74535580	8.472690	TRUE	742	6283	4661986	8.467655	4

4.3.8 PAIRS.RDATA

Description

This output file has the combined pairs table of final trades for all of the markets, appending together the pairs tables from each NAICS market. The firm synthesis model component produces these outputs.

Field Definitions

Table 98 describes the fields in the Pairs.Rdata file, while Table 99 shows a snapshot of one of the output files.

TABLE 98: FORMAT OF PAIRS.RDATA

FIELD	UNIT	DESCRIPTION
BuyerID	-	A sequential ID numbers for the buying firm (the firms IDs range from 1 to N where N is the total number of firms synthesized by the model)
SellerID	-	A sequential ID numbers for the selling firm (the firms IDs range from 1 to N where N is the total number of firms synthesized by the model)

FIELD	UNIT	DESCRIPTION
Production_zone	-	Production zone, the zone where the selling firm is located, using the Mesozone system
Consumption_zone	-	Consumption zone, the zone where the buying firm is located, using the Mesozone system
NAICS	-	NAICS code of the selling firm and the commodity being traded, using the 6-digit US BEA system as used for the IO tables
Seller.NAICS	-	NAICS code of the seller firm, using the 6-digit US BEA system as used for the IO tables
Commodity_SCTG	Employee	Standard Classification for Transported Goods, for the commodity being traded
Seller.Size	Pounds	Size of the selling firm in number of employees (which represents the mid-points of firm size ranges from the county business patterns data)
OutputCapacityTons	-	Annual Output Capacity in POUNDS of the selling firm (Note that we are working in POUNDS and not tons -- changed to avoid rounding)
Buyer.NAICS	-	NAICS code of the buying firm, using the 6-digit US BEA system as used for the IO tables
Buyer.SCTG	Employee	Standard Classification for Transported Goods, for the commodity produced by the buying firm; zero for firm that does not produce any transported goods
Buyer.Size	$	Size of the buying firm in number of employees (which represent the mid-points of firm size ranges from the county business patterns data)
ConVal	Pounds	Consumption value: Annual value in $ of the commodity being traded required by the buying firm to satisfy their production requirements
PurchaseAmountTons	Miles	Annual purchase amount of the commodity being traded in POUNDS required by the buying firm to satisfy their production requirements
Distance	-	Great circle distance (GCD) in miles between the selling and buying firms
distchannel	-	Distribution channel assigned to this pair of firms by the distribution channel model (1=direct, 2=1 stop/transfer, 3=2 stops/transfers, 4=3 stops/transfers)
ship_size	Pounds	Shipment size category of the individual shipment between this seller and buyer (1=<1000 lbs, 2=1000 to 9999 lbs, 3=10000 or more pounds)
weight	-	Shipment size of the individual shipments between this seller and buyer in pounds
lssbd	$	Flag for large seller, small buyer, and distance of more than 300 miles between them

FIELD	UNIT	DESCRIPTION
MinGmnql	-	Logistics cost for minimum path
MinPath	Days	Minimum path assigned by the mode choice model
Attribute2_ShipTime	$/Pound	Shipment travel time from seller to buyer in days
Attribute1_UnitCost	Pounds	Unit cost of shipment transportation
Quantity.Traded	-	The quantity in POUNDS traded between this seller and buyer in this iteration of the PMGs (note the table has been filtered down to just the records with a trade a in the final iteration so the value in this field is the same as the Last.Iteration.Quantity
Number.of.Trades	Pounds	The number of trades between this seller and buyer
Last.Iteration.Quantity	Pounds	The quantity in POUNDS traded between this seller and buyer in the final iteration of the PMGs

TABLE 99: SNAPSHOT OF PAIRS.RDATA

	BuyerID	SellerID	Production_zone	Consumption_zone	NAICS	Seller.NAICS	Commodity_SCTG	Seller.Size	OutputCapacityTons	Buyer.NAICS	Buyer.SCTG	Buyer.Size
1:	2	7373544	385	151	113000	113000	25	794	6736398051	113000	25	10
2:	9	6381951	205	153	113000	113000	25	60	509275347	113000	25	10
3:	16	6381951	205	153	113000	113000	25	60	509275347	113000	25	10
4:	23	6382147	272	153	113000	113000	25	60	509275347	113000	25	10
5:	23	7373544	385	153	113000	113000	25	794	6736398051	113000	25	10
6:	30	7472	265	152	113000	113000	25	10	84879225	113000	25	10

4.3.9 PAIRS.CSV

Description

This output file is the combined pairs table of final trades for all of the markets, appending together the pairs tables from each NAICS market. The firm synthesis model component produces this output.

Field Definitions

For the table format and snapshot, please see Table 98 and Table 99 above under the "PAIRS.RDATA" section.

4.3.10 PMG_WORKSPACE.RDATA

Description

This output file has the workspace containing function, variables and other data from the early portion of the R model that is used in each of the separate R script processes to support development of the inputs to the PMGs. The firm synthesis model component produces these outputs.

4.3.11 WAREHOUSES.RDATA

Description

This output file has the list of warehouse locations summarized from the firm synthesis for use in the warehouse allocation model. The firm synthesis model component produces these outputs.

Field Definitions

Table 100 describes the fields in the Warehouses.Rdata file. Table 101 is a snapshot of one of the output files.

TABLE 100: FORMAT OF WAREHOUSES.RDATA

FIELD	UNIT	DESCRIPTION
CBPZONE	-	County Business Pattern (CBP) zone
FAFZONE	-	FAF zone
Industry_NAICS6_Make	-	Six-digit NAICS (BEA) code of the industry
Commodity_SCTG	-	SCTG two-digit code of the commodity
esizecat	-	Employment size category
Emp	Employee	Number of employees
BusID	-	Business ID number
MESOZONE	-	Mesozone ID number

TABLE 101: SNAPSHOT OF WAREHOUSES.RDATA

	CBPZONE	FAFZONE	Industry_NAICS6_Make	Commodity_SCTG	esizecat	Emp	BusID	MESOZONE
1:	17031	171	481000	0	1	10	2181012	19
2:	17031	171	481000	0	1	10	2181013	23
3:	17031	171	481000	0	1	10	2181014	4
4:	17031	171	481000	0	1	10	2181015	2
5:	17031	171	481000	0	1	10	2181016	26
6:	17031	171	481000	0	1	10	2181017	21

4.3.12 VEHTOURPAT_ALLCOMMODITIES.CSV

Description

This output file shows the frequency of each alternative (combination of tour patterns and vehicle types) for all commodities. The vehicle choice and tour pattern model component produces this output.

Field Definitions

Table 102 describes the fields in the vehtourpat_allcommodities.csv file. Table 103 is a snapshot of this file.

TABLE 102: FORMAT OF VEHTOURPAT_ALLCOMMODITIES.CSV

FIELD	UNIT	DESCRIPTION
Var1	-	Alternative number
TP_Veh	-	Tour pattern and vehicle type combination
Freq	-	Frequency

TABLE 103: SNAPSHOT OF VEHTOURPAT_ALLCOMMODITIES.CSV

	A	B	C
1	Var1	TP_Veh	Freq
2	1	direct, 2 axle	150185
3	2	direct, 3-4 axle	40906
4	3	direct, semi/trailer	70471
5	4	peddling, 2 axle	67316
6	5	peddling, 3-4 axle	18046
7	6	peddling, semi/trailer	32290

4.3.13 VEHTOURPAT_FOOD.CSV

Description

This output file shows the frequency of each alternative (combination of tour patterns and vehicle types) for shipments of food products. The vehicle choice and tour pattern model component produces this output.

Field Definitions

Table 104 describes the fields in the vehtourpat_food.csv file. Table 105 is a snapshot of this file.

TABLE 104: FORMAT OF VEHTOURPAT_FOOD.CSV

FIELD	UNIT	DESCRIPTION
Var1	-	Alternative number
TP_Veh	-	Tour pattern and vehicle type combination
Freq	-	Frequency

TABLE 105: SNAPSHOT OF VEHTOURPAT_FOOD.CSV

Var1	TP_Veh	Freq
1	direct, 2 axle	32666
2	direct, 3-4 axle	5052
3	direct, semi/trailer	7992
4	peddling, 2 axle	25295
5	peddling, 3-4 axle	4227
6	peddling, semi/trailer	6971

4.3.14 VEHTOURPAT_MFG.CSV

Description

This output file shows the frequency of each alternative (combination of tour patterns and vehicle types) for manufactured products. The vehicle choice and tour pattern model component produces this output.

Field Definitions

Table 106 describes the fields in the vehtourpat_mfg.csv file, while Table 107 shows a snapshot of the output files.

TABLE 106: FORMAT OF VEHTOURPAT_MFG.CSV

FIELD	UNIT	DESCRIPTION
Var1	-	Alternative number
TP_Veh	-	Tour pattern and vehicle type combination
Freq	-	Frequency

TABLE 107: SNAPSHOT OF VEHTOURPAT_MFG.CSV

Var1	TP_Veh	Freq
1	direct, 2 axle	43112
2	direct, 3-4 axle	10087
3	direct, semi/trailer	20770
4	peddling, 2 axle	17683
5	peddling, 3-4 axle	4201
6	peddling, semi/trailer	9629

4.3.15 STOPSEQ_NUMSTOPSPERTOUR.CSV

Description

This output file shows the frequency of each stop count category. The stop sequence model component produces this output.

Field Definitions

Table 108 describes the fields in the stopseq_numstopspertour.csv file. Table 109 is a snapshot of this file.

TABLE 108: FORMAT OF STOPSEQ_NUMSTOPSPERTOUR.CSV

FIELD	UNIT	DESCRIPTION
stop_count	-	Stop count category
Frequency	-	Frequency

TABLE 109: SNAPSHOT OF STOPSEQ_NUMSTOPSPERTOUR.CSV

	A	B
1	stop_count	Frequency
2	1	5410
3	2	2403
4	3	1506
5	4	1151
6	5	753
7	6	650
8	7	515
9	8	372
10	9	320
11	10	260
12	11	255
13	12	191
14	13	157
15	14	185
16	15	152
17	16	165
18	17	122
19	18	113
20	19	92
21	20 or more	1592

4.3.16 STOPSEQ_TOURCATBYSHIPSIZE.CSV

Description

This output file shows the frequency of tour category by shipment size. The stop sequence model component produces this output, summarizing the output from the number of tours model.

Field Definitions

Table 110 describes the fields in the stopseq_tourcatbyshipsize.csv file, while Table 111 shows a snapshot of the output files.

TABLE 110: FORMAT OF STOPSEQ_TOURCATBYSHIPSIZE.CSV

FIELD	UNIT	DESCRIPTION
Tour_category	-	Tour category
< 999 lbs	-	Shipment size category 1
1k-10k lbs	-	Shipment size category 2
> 10k lbs	-	Shipment size category 3

TABLE 111: SNAPSHOT OF STOPSEQ_TOURCATBYSHIPSIZE.CSV

	A	B	C	D
1	Tour_category	< 999 lbs	1k-10k lbs	> 10k lbs
2	All stops in 1 tour	213052	13744	2350
3	All stops in 2 tours	72240	4974	747
4	All stops in 3 tours	45611	4804	914
5	All stops in 4 tours	16584	3014	1180

4.3.17 STOPSEQ_STOPCLUSTER_SAMPLE.PNG

Description

As shown below in Figure 18, this output file shows a chart of stop clusters for a sample of patterns from each tour type category. The stop sequence model component produces this output based on the results of the clustering model.

Field Definitions

Figure 18 shows a snapshot of the output files. This is a png file graphic including x_coord, y_coord and Tour_ID.

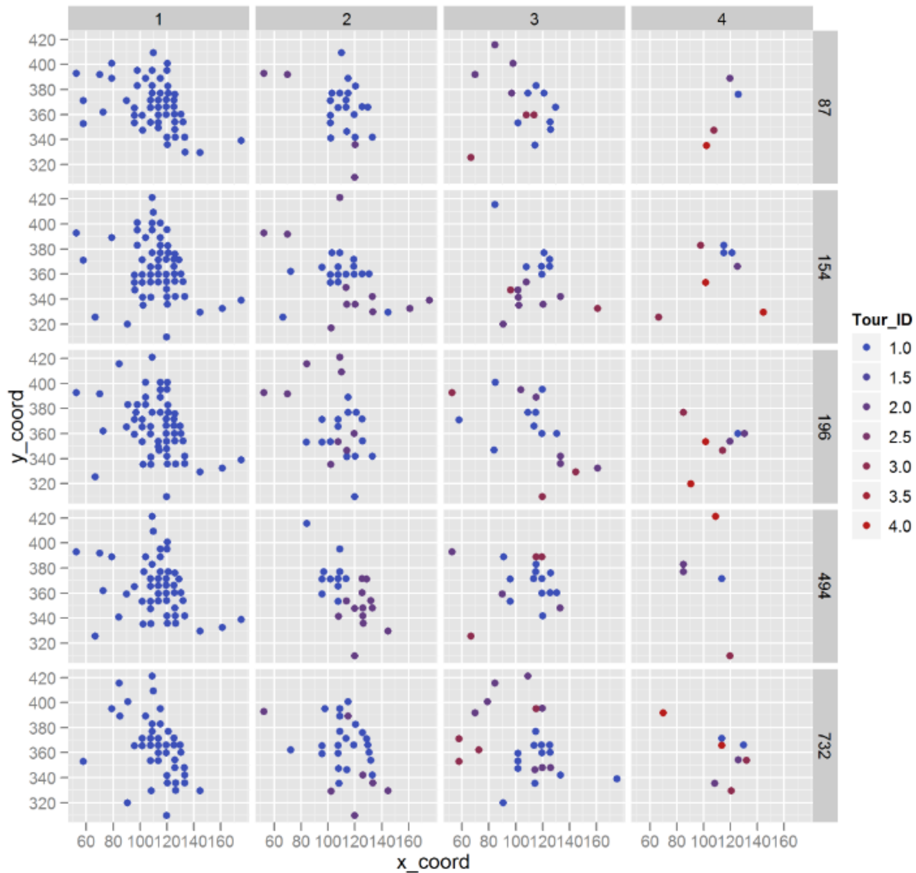

FIGURE 18: SNAPSHOT OF STOPSEQ_STOPCLUSTER_SAMPLE.PNG

4.3.18 STOPDUR_DURATIONBYTOURTYPE.CSV

Description

This output file shows the frequency of stop duration categories by tour types. The stop duration model component produces this output.

Field Definitions

Table 112 describes the fields in the stopdur_durationbytourtype.csv file, while Table 113 shows a snapshot of the output files.

TABLE 112: FORMAT OF STOPDUR_DURATIONBYTOURTYPE.CSV

FIELD	UNIT	DESCRIPTION
Stop_Duration	-	Stop duration category
DirectTour_Stops	-	Frequency of direct tour stops
PeddlingTour_Stops	-	Frequency of peddling tour stops

TABLE 113: SNAPSHOT OF STOPDUR_DURATIONBYTOURTYPE.CSV

	A	B	C
1	Stop_Duration	DirectTour_Stops	PeddlingTour_Stops
2	< 15 mins	1720	43868
3	15-30 mins	6574	34517
4	30-45 mins	11609	13030
5	45-60 mins	17831	8603
6	60-75 mins	14376	5580
7	> 75 mins	206194	15312

4.3.19 TOD_TODBYTOURTYPE.CSV

Description

This output file shows the frequency of stop duration categories by tour types. The tour time of day model component produces this output.

Field Definitions

Table 114 describes the fields in the tod_todbytourtype.csv file, while Table 115 shows a snapshot of the output files.

TABLE 114: FORMAT OF TOD_TODBYTOURTYPE.CSV

FIELD	UNIT	DESCRIPTION
TOD	-	Time of Day category
Direct_Tours	-	Frequency of direct tour stops
Peddling_Tours	-	Frequency of peddling tour stops

TABLE 115: SNAPSHOT OF TOD_TODBYTOURTYPE.CSV

	A	B	C
1	TOD	Direct_Tours	Peddling_Tours
2	Before 6 AM	987	12853
3	6-8 AM	10803	1527
4	8-9 AM	19299	1399
5	9-10 AM	20581	310
6	After 10 AM	69153	275

4.3.20 TRIP_TABLE.CSV

Description

This output file shows the number of trips by origin mesozone, destination mesozone, vehicle type, and time of day. The trip table model component produces this output.

Field Definitions

Table 116 describes the fields in the trip_table.csv file, while Table 117 shows a snapshot of the output files.

TABLE 116: FORMAT OF TOD_ TRIP_TABLE.CSV

FIELD	UNIT	DESCRIPTION
ORI_mz	-	origin mesozone
DES_mz	-	destination mesozone
Vehicle	-	vehicle type
TimeOfDay	-	time of day
V1	-	Number of trips

TABLE 117: SNAPSHOT OF TRIP_TABLE.CSV

	A	B	C	D	E
1	ORI_mz	DES_mz	Vehicle	TimeOfDay	V1
2	1	1	2-axle	MD	4
3	1	1	2-axle	AM	2
4	1	1	semi/trailer	MD	1
5	1	1	semi/trailer	AM	1
6	1	1	3,4 axle	MD	1
7	1	1	3,4 axle	AM	2
8	1	2	2-axle	AM	5

5.0 DEVELOPING ALTERNATIVE SCENARIOS

The basic setup of the Chicago Regional Meso-Scale Freight Forecasting Model with Agent-Based Supply Chain Evolution is described above in Sections 3.0 (structural relationships) and 4.0 (data files). To develop alternative scenarios, the user needs to consider which aspects of model behavior to vary and which aspects to hold constant. Four groups of input parameters govern different levels of the model system:

- Macroeconomic conditions – employment, productivity, consumption, imports/exports (Section 3.2)
- Business/operating strategies – firm synthesis attribute assumptions (Section 3.3)
- Transport and logistics costs and capacities (Section 4.2)
- Microeconomic interactions – agent-based behavioral parameters (Section 3.6)

Definitions of the data tables and R scripts for scenario inputs appear in Sections 4.1-0, respectively. The remainder of this section discusses several types of scenario alternatives in terms of which parameters to change, how and where (files) to make the changes, and any relevant assumptions that might affect the validity of these changes.

5.1 BASE-YEAR SCENARIO

Before discussing the creation of alternative scenarios, this section summarizes the assumptions inherent in the base-year scenario as a set of starting conditions. The base-year scenario assumes the economic conditions, industry linkages, and logistic practices that produced the input files to the meso-scale model, described above in Section 3.0:

5.1.1 DOMESTIC PRODUCTION

- For each commodity, the rate of domestic production per employee is a function of the "observed" amount produced in the base year, divided by total domestic employment in the base year for the same commodity-producing industry.
    - The total amount produced is derived from the BEA IO tables, factored to represent 2010 conditions (data_2010io.csv).
    - The total domestic employment is derived from the County Business Patterns control totals (data_emp_cbp.csv).
    - All domestic firms in the same NAICS code have the same per-employee production rates, irrespective of region or firm size.

5.1.2 DOMESTIC CONSUMPTION

- For each commodity produced, the dollar-valued quantities of various input commodities consumed are proportional to the amounts shown in the BEA IO tables, which reflect the manufacturing process relationships (input requirements) found in the 2010 base year (data_2010io.csv).

5.1.3 LOCAL EMPLOYMENT MIX

- The allocation of employment by industry type to the meso zones within the greater Chicago portion of the modeling system depends on the base year mix of employment by industry (data_mesozone_emprankings.csv).

5.1.4 FOREIGN PRODUCTION

- For each commodity, the U.S. import dollar value represents an upper-bound on the production capacity of firms in each represented country, as derived from USA Census Trade Online for the base year (data_foreign_prod.csv).
- Import quantities by NAICS code and country are assumed contemporaneous with domestic production for these same NAICS codes, thus foreign and domestic producers compete for the same domestic consumers in procurement markets.
- Trade between non-U.S. countries does not affect the production of goods for sale to the U.S.

5.1.5 FOREIGN CONSUMPTION

- For each commodity, the U.S. export dollar value represents an upper-bound on the consumption demands of firms in each represented country, as derived from USA Census Trade Online for the base year (data_foreign_cons.csv).
- Export quantities by NAICS code and country are assumed contemporaneous with domestic consumption for these same NAICS codes, thus foreign and domestic buyers compete for the same domestic sellers in the procurement markets.
- Trade between non-U.S. countries does not affect the demand for U.S. goods from other countries.

5.1.6 PRODUCER PRICES

- Unit prices of goods are based on the FAF3 stratification by 43 SCTG categories (data_unitcost.csv).
- NAICS 6-digit groups are mapped onto these SCTG categories.
- For a given SCTG group, producer unit prices are the same for all domestic manufacturers, irrespective of region or firm size.
    - Domestic wholesalers charge the domestic producer price plus a 20% markup, irrespective of commodity type or firm size.
    - Foreign manufacturers charge the domestic producer price minus a 10% discount, irrespective of country or commodity type.
- Prices offered by agents in the buyer-seller matching game, and general trade outcomes, do not affect the dollar-valued production or consumption rates found in the IO tables.

5.1.7 BUSINESS OPERATING STRATEGIES

- Based on the logistics literature, businesses are assumed to differ in the strengths of their preferences for costs savings versus responsive service. Commodities generally viewed as "functional" are considered to have a low value-to-weight ratio, and thus cost savings is proportionally more important than responsiveness. Commodities that are generally viewed as "innovative" are generally considered to have a high value-to-weight ratio, and thus responsiveness is proportionally more important than thus cost savings.

- For similar reason, it is assumed that buyers attempt to minimize risk by spreading their sourcing contracts to multiple suppliers, more so for innovative commodities.
- For each commodity group defined by the SCTG mapping, data_firm_pref_weights.csv indicates whether the commodity is considered functional, semi-functional, semi-innovative or innovative, and assigns the proportional preference weights and sourcing constraints shown below in Table 118 during the firm synthesis step.

TABLE 118: BUYER PREFERENCE WEIGHTS AND CONSTRAINTS BY COMMODITY TYPE

COMMODITY_TYPE	COSTWEIGHT	TIMEWEIGHT	SINGLESOURCEMAXFRACTION
Functional	0.8	0.2	1.0
Semi-Functional	0.6	0.4	0.9
Semi-Innovative	0.4	0.6	0.9
Innovative	0.2	0.8	0.8

5.1.8 TRANSPORT AND LOGISTICS COSTS

- The parameters estimated for use in the transport and logistics cost function represent base-year logistics practices (service rates) and cost structures, as specified in the file, scenario_variables.R. These assumed rates include:
  - Fees
    - Weight-based handling fees for intermodal lifts, transloading, and air cargo
    - Line-haul charges per ton-mile by mode (water, rail car, intermodal, air) and truck vehicle (4 options)
  - Service attributes
    - Handling times for intermodal lifts, transloading, and air cargo
    - Weight capacities for trucks, rail cars, and airplanes
    - Discount rates for goods carrying costs (stratified by 3 commodity groups)
    - Variability in demand flow rates (stratified by 3 commodity groups)
    - Safety stock constants (stratified by 3 commodity groups)
- The transport and logistics cost function is assumed to be transferable for all shipments of the same commodity type, irrespective of firm attributes, or shipment origin or destination.
- Mode-specific travel times are derived from the skims file, which is itself derived from network least-cost path calculations for each mode used along the path, adding in the above-mentioned assumed transshipment node times.

- As currently structured, travel times and transshipment times are static and are not capacity constrained.
- Transshipment nodes have the same service rate characteristics regardless of size or geographic location.

5.1.9 AGENT BEHAVIORAL PARAMETERS

- PMG behavioral parameters reflect assumptions about the possible mindsets of buying and selling agents as they seek out and try to secure favorable procurement contracts for their firms. (See Section 3.6 for theoretical propositions.)
- The rank ordering of PMG parameters applied to the transport cost utilities are grounded in theoretical propositions taken from game theory. Under the initial setup, they are assumed to apply to all commodity markets being run during the same simulation.
- A baseline set of PMG parameters are recommended, motivated by the aspiration to represent plausible agent behavior under the following assumptions:

  1. Agents do not have perfect information about all agents in the market, but do have information about the agents with whom they have had trading encounters and know both their own rating of past trades as well as how their trading partners rated them.
     - ClairvoyantInitialExpectedPayoffs = 0 (false) Perfect information on what all potential trading partners have to offer before trading begins is unrealistic.
  2. Agents learn through successive iterations, accumulating experiences and updating their beliefs about agents with whom they have new trading encounters.
     - DynamicAlternatePayoffs = 1 (true) Recalculating alternate payoffs every iteration based on updated expected payoffs is consistent with the notion of learning from experience and knowledge acquisition.
  3. Both buyers and sellers evaluate the potential for future trades with each other using weighted payoff values that consider the perceived strength of trading relationships, as opposed to pure cost minimization or revenue maximization.
     - SellersRankOffersByOrderSize = 0 (false) If false, invokes a preference for sellers ranking buyers based on expected payoffs, which are closely correlated with order size but are weighted by evaluations of past trading experiences.
  4. The commodity market under study is in a steady state, and agents are price-takers who do not affect average commodity prices in the short run.

     Alternative "A": Under the assumption that modeled costs are unbiased and accurately represent average conditions, the objective of the scenario is to simulate a set of commodity flows that is consistent with these assumptions.

     - Payoff Matrix (see Figure 15: Mutual knowledge without bargaining): Assumes agents are price-takers and do not influence short-run commodity prices. Agents know how each rates the other after a trade encounter.

- BothCoop = 1.0
- Temptation = 0.7 (arbitrarily lower than BothCoop; Temptation = Sucker = BothDefect)
- Sucker = 0.7
- BothDefect = 0.7

Alternative "B": Although agents do not affect *average* commodity prices in the short run, commodity prices will vary across individual trading pair. Buyers and sellers bargain, deviating from the average generalized costs in proportion to successive payoff-weight adjustments over iterative game play. Under the assumption that modeled costs are unbiased, the objective of the scenario is to simulate a set of commodity flows patterns that reflect local variation in commodity prices.

- Payoff Matrix (See above Figure 16: Mutual knowledge with bargaining) Assumes that payoff weights represent adjustments to nominal (average) costs and revenues.
  - BothCoop = 1.0
  - Temptation = 1.3
  - Sucker = 0.6
  - BothDefect = 0.7

5. It is unknown whether buyers are of the type who tend to satisfice quickly and stick with their initial trading partners, or are of the type who are inclined to seek out potentially new and better trading partners.
   - Exogenous Payoff Values:
     - InitExpPayoff = BothCoop (neutral) Encourages neither inertia nor experimentation.
     - RefusalPayoff = 0.5 (arbitrarily lower than lowest payoff weight)
     - WallflowerPayoff = 0.0 Does not affect game, but would eliminate sellers who do not trade

5.2 GROWTH SCENARIO ASSUMING BASELINE TRENDS CONTINUE

A future-year scenario that assumes the continuation of base-year trends, but accounts for the expected growth in employment would seem to be straight forward, but is more involved than simply factoring employment. A number of steps need to take place in order to ensure that the continuation of existing trends is reflected:

1. The County Business Patterns table (data_emp_cbp.csv) that is used as a control total for U.S. employment and firm size distributions needs to be factored to match the expected growth in employment by industry sector in order for the firm synthesis step to produce the right number and size of firms. Without additional information, it could be assumed that the size distribution of firms in each industry group could remain proportionally the same for the future year.

Growth in employment outside of the Chicago region will likely come from an external data source, which could be geographically non-specific, such as BEA national forecasts of growth by industry sector. Geographically specific forecast (e.g., by state or county) are typically available from commercial providers (e.g., IHS, Woods & Poole).

2. The BEA IO table (data_2010io.csv) should be adjusted such that the total amounts produced and consumed by each industry is proportional to the expected growth in employment in that industry in order to keep production rates per employee the same as in the base year. For example, if the assumption is 20% growth in manufacturing sectors and 30% growth in other sectors, then the row totals in the IO table that correspond to manufacturing NAICS codes would be factored by 20% and the row totals corresponding to other sectors would be factored by 30%. Additionally, the column totals for these manufacturing and non-manufacturing sectors would be factored by 20% and 30%, respectively.

   The next step is to apply iterative proportional fitting (IPF) to the table to derive new cell values.[12] The factored-up row and column totals become marginal target values, and the original IO table cell values become the seed matrix. In the IO table, total production must equal total consumption (the sum of all row totals should equal the sum of all column totals). Keeping with the objective to maintain constant production rates, the column marginal target values should be rescaled by the ratio of the sum of all row totals (total production) divided by the sum of all column totals (total consumption). That some industries are growing by a different percentage than others implicitly imposes a structural change on the input consumption relationships between industries, a mathematical artifact that is independent of any theorized technological changes in production.

3. In addition to domestic employment, it is necessary to show proportional growth in imports and exports by industry type. Assuming the objective is a continuation of existing trends, the growth in imported commodities to be consumed by U.S. firms should be proportional to the expected growth in U.S. firms that use these same commodities. With reference to the base year, the percentage change in the IO table column totals (consumption) for each commodity (from Step 2) should be applied to the total amounts produced by each country of that same commodity in data_foreign_prod.csv. Similarly, the percentage change in the IO table row totals (production) for each commodity should be applied to the total amounts exported to each foreign country in data_foreign_cons.csv.

4. Within the Chicago region meso zones, assumed growth in employment may change the spatial distribution of jobs within the region. For example, a meso zone that was somewhat rural in the base year, may be projected to gain manufacturing employment in the growth scenario and thus its ranking in manufacturing jobs may increase to a higher percentile. In such cases, the table found in data_mesozone_emprankings.csv should be modified to reflect these changes.

5.3 MACROECONOMIC VARIATIONS FROM CONTINUATION OF TRENDS

The growth scenario with "trend continuation" is an important starting point for considering variations to inputs and assumptions in order to create different scenarios. Below are some plausible macroeconomic

[12] The meso-scale freight model does not have a built in iterative proportional fitting (IPF) process. It is assumed that the user would perform this task outside of the program.

variations that could be considered, either individually or in combination, and the steps required to implement these changes.

5.3.1 CHANGES TO INDUSTRIAL PRODUCTIVITY

Changes in industrial productivity are a realistic consideration for any future-year forecasting scenario. Certain industrial sectors, particularly manufacturing and technology, have historically produced more output per unit input over time. For example, over the past decade U.S. output of microelectronics has grown steadily while employment in this industry has remained relatively flat. Thus, the amount of production per employee will tend to increase over time and failure to account for productivity increases will likely lead to under prediction of commodity flows and freight movements.

The U.S. Bureau of Labor Statistics' Multi-Factor Productivity Index (http://www.bls.gov/mfp/) provides a set of indices for various industries that show the changes in output per change in input over more than two decades. These trend lines could be extrapolated to predict future productivity growth by industrial sector.

1. Derive growth in multi-factor productivity (MFP) for each NAICS commodity sector for the forecast year.
2. Starting with the adjusted BEA IO table from the "trend continuation scenario," apply the derived MFP factors from Step 1 to the row marginal target values as an adjustment to create new row target values.
3. Repeat the IPF process described above, using the adjusted row target values and creating new column target values. The expected result should be an increase production and consumption of commodities, compared with the "trend continuation scenario" while holding domestic employment the same.
4. In order to reflect these changes in the import and export demand, repeat Step 3 of the "trend continuation scenario."

5.3.2 CHANGES TO DOMESTIC CONSUMPTION

As discussed above in Sections 5.1.1 and 5.1.2, changes to employment and production and consumption rates are intertwined; therefore, changing one of these three dimensions will and should have an impact on the other two dimensions and, possibly, on exports and imports. As described above, the IPF procedures are production-constrained in order to preserve the relationship between employment levels and output. Theoretically, it would be possible to change domestic consumption levels independently and create a consumption-constrained version of the IO table using an IPF process. This would have the effect of changing production outputs. Holding employment constant, this would imply changes in productivity per employee. Alternatively, constraining production levels to consumption levels and holding output per employee constant, the user could justify changes in employment in a particular industry or group of industries. For example, substitution of clean-energy technologies could reduce the consumption of fossil fuels. Due to the complexity of these interdependent processes, consumption-driven changes to the IO table are not recommended.

5.3.3 CHANGES TO EXPORTS AND IMPORTS

Scenarios in which the analyst may want to deviate from the baseline rates of imports and exports may include the following examples:

- Global economic growth or decline resulting in a change in the level of demand across all commodity sectors and all regions;
- Change in demand for a particular commodity, or set of commodities, (e.g. luxury goods, agricultural products, petroleum) across all regions;
- Change in demand in a particular country or region of the world, or set of countries (e.g., China, European Union, OPEC, NAFTA) across all commodities; and
- Changes to demand of some subset of commodities intersected with a subset of regions.

As describe above in in Sections 5.1.4 and 5.1.5, the user can make these changes by directly manipulating the appropriate foreign demand files and factoring up or down the values by the assumed change in demand:

- Exports: data_foreign_cons.csv
- Imports: data_foreign_prod.csv

As shown in Sections 4.1.7 and 4.1.8, both files have NAICS codes for commodities and are broken down by country. If so desired, it should be relatively easy to group entries by commodity group (e.g., Agriculture) or country group (e.g. European Union) and apply percentage change factors uniformly within each group; however, this would require some additional work to create group memberships, which has not been done as part of the initial model deliverable.

5.3.4 CHANGES TO COMMODITY PRICES

It may be desirable to assert an assumption about commodity price changes, such as crude petroleum or agricultural goods. Changes to commodity prices also may be motivated by the objective of achieving more variation across market segments; for example:

- Geographic variation across regions of the U.S. or across countries;
- Firm size variation, assuming some correlation with prices exist;
- Industrial variation, breaking down the SCTG categories into NAICS categories.

Note that these market segmentation schemes require additional information that was not available when creating the base-year model, but may be available or derived in the future.

For all of these cases, the simplest way to affect these changes is to modify the unit cost input file (data_unitcost.csv), which is essentially a lookup table. For simple changes without additional segmentation, table values may be changed for the appropriate commodity group. If additional segmentation is desired, then both the table and the R script that reads it and assigns unit costs when creating sellers should be modified to use additional columns related to the segmentation variables of interest (geographic identifiers, firm size identifiers, etc.).

5.4 CHANGES TO TRANSPORT AND LOGISTICS COSTS

Changes to transport and logistics cost parameters may be implemented in a relatively straightforward manner by changing the values described above in Section 5.1.8. For example, one might want to consider the impact of a policy that would reduce variability in demand flow rates or stock-out probabilities. As another example, changes in fuel prices for specific modes would be reflected here by changing the values of line-haul costs.

Most of the changes could be made in the script, scenario_variables.R. Changes to network connectivity and facility capacity would need to be made directly in the freight model network files and new skims produced.

5.5 CHANGES TO BUSINESS OPERATING STRATEGIES

Changes to business operating strategies, the assumed preferences of buyer and seller agents in the PMG, may be made in the file, data_firm_pref_weights.csv, as discussed above in Section 5.1.7. As currently formulated, this particular file is very flexible and simply keys on SCTG commodity groups. The four groupings labeled according to the functional-vs-innovative continuum logic are there for convenience.

Mechanically, it is possible for every SCTG group to have a unique set of preference weights on shipping time and cost and on sourcing maximum fraction. The only restriction is that the preference weights must sum to 1.0. SingleSourceMaxFraction should be a value greater than 0 and not exceeding 1.0 and, in practice, it has been found that smaller fractions (e.g., less than 0.50) tend to create conditions that are impossible to satisfy for many markets in which the ratio of buyers to sellers is high, leading to lengthy run times. In theory, it also would be possible to stratify this file further to key on the more detailed NAICS codes; however, this would require changes to the R script for firm synthesis.

5.6 CHANGES TO AGENT MARKET BEHAVIOR

Changes to procurement market behavior, the matching of spatially distributed buyers and sellers, which determines spatial flow patterns, can be made directly in the section of scenario_variables.R that specifies the PMG .ini file. While Section5.1.9 recommends some conservative starting assumptions, Sections 3.6.6 and 3.6.10 suggest different ways to configure game play to reflect other market assumptions.

APPENDIX A. ROAD MAP FOR C++ CODE

Header and code files are stored in the same source directory. The complete source code distribution includes three general types of files:

- PMG Classes and Functions;
- TNG Classes and Functions; and
- Utility Classes and Functions.

Five new source/header file pairs were created for the PMG program. The files with names in all caps are TNG distribution files that are virtually unchanged from the distribution originals. Additionally, the public license **wingetopt.c** and **wingetopt.h** files are unchanged.

PMG Classes and Functions

- **pmg.h & pmg.cpp –** The main program. Instantiates, initializes, and runs the simulation object.
- **pmgsim.h & pmgsim.cpp –** The main simulation object (subclass of TNG bioSimulation class) that implements the simulation's match/trade loop structure.
- **pmgbot.h & pmgbot.cpp –** Class (pmgTradeBot) representing an individual buyer or seller. This class implements the trading strategies of the buyers and sellers based on the objectives defined in their input parameter files.
- **pmgpop.h & pmgpop.cpp –** Class that represent a population of buyer or seller Tradebots. This is a legacy class not particularly important, as we are not using the genetic algorithm features of TNG.
- **attrib.h & attrib.cpp –** Utility classes for reading the buyer, seller, and costs input files that define the buyer and seller Tradebots attributes and objectives.

TNG Classes and Functions[13]

- **SBSYS.H** Header file that includes all SimBioSys header files
- **SIMBIO.H & SIMBIO.CPP** Generic bioObject class definition serves as base class for other TNG classes.
- **WORLD.H & WORLD.CPP** Abstract base class (bioWorld) is responsible for the physics governing the virtual environment of the simulation
- **SIMULAT.H & SIMULAT.CPP** Abstract base class, bioSimulation. This class contains member functions and data for the construction of a world and the populations of agents that inhabit the world.
- **THING.H & THING.CPP** Abstract base class (bioThing) represents the inhabitants of the world. These inhabitants are either passive entities or active autonomous agents. The bioThing class identifies certain general operations common to all inhabitants and provides for the storage and retrieval of the current positions and orientations of the inhabitants.

---

[13] Some of the TNG class descriptions are summarized from "A Computational Laboratory for Evolutionary Trade Networks" TNG documentation at http://www2.econ.iastate.edu/tesfatsi/tngieee.pdf.

- **POPULAT.H & POPULAT.CPP** Abstract base class (bioPopulation) that defines data and operations required for the initial construction and reproduction of the agent populations.
- **AGENT.H & AGENT.CPP** An abstract base class, bioAgent, is a derived bioThing class that represents the subset of world inhabitants who are agents. This class sets general protocols for communication and interactions among agents and for interactions between agents and passive entities.
- **GTYPE.H & GTYPE.CPP** An abstract base class, bioGType, identifies the basic elitism, recombination, and mutation operations used in the genetic reproduction of agent populations. The evolutionary genetic functionality of TNG is not used by PMG but the source file is included as it forms part of the TNG distribution and the legacy class hierarchy.
- **PTYPE.H & PTYPE.CPP** Class (bioPType) derived from bioAgent, bioPType, stores an instance of bioGType that is used by bioPopulation to construct an agent's program before the agent is added to the world. The phenotype functionality of TNG is not used by PMG.
- **CELWORLD.H & CELWORLD.CPP** Subclass of bioWorld not used by PMG
- **PROGRAM.H** Abstract class (bioProgram) sets general protocols for the communication between an agent and its program. Provides the ability to substitute different program implementations, such as finite-state machines (FSMs), artificial neural networks, and Turing machines, without changing any other aspect of SimBioSys. (This functionality is not used by PMG.)
- **NNET.H & NNET.CPP** Legacy TNG Framework neural network bioProgram subclass not used by PMG
- **FSM.H & FSM.CPP** Legacy TNG Framework Finite State Machine bioProgram subclass not used by PMG

Utility Classes and Functions

- **LIST.H & LIST.CPP** Class (bioList) for maintaining and modifying lists of bioObjects (e.g. Append, Remove, Count)
- **INIFILE.H & INIFILE.CPP** Utility routines for reading ini file parameter values
- **QSORT.C** Public domain non-recursive C language quicksort function.
- **wingetopt.h & wingetopt.c** Public domain C language utility to read Windows command line parmeters.
- **RANDOM.H & RANDOM.C** C language random number generator functions
- **BITVECTOR.H & BITVECTOR.CPP** Class (bioBitVector) providing bit vector with underlying storage as packed integer array.
- **XMATRIX.H & XMATRIX.CPP** Legacy TNG matrix manipulation utility class not used by PMG

APPENDIX B. rFREIGHT^{TM} R PACKAGE DOCUMENTATION