Scenario Modeling Tool
Data Science & Innovation

Abstract

An exercise in prescriptive analytics, the project aims to enable Cisco to evaluate changes and variations to their global supply chain, and quickly realize their impact on product routing, warehousing network and use, transportation mode mix, cost, customer satisfaction, and the environment.

Contents

1 What is Scenario Modeling?

A scenario represents a specific state of the global supply chain. Scenarios may arise in practice, or as exercises in risk management and strategic planning.

2 Supply Chain Overview

Daily operations are executed by a network of contract manufacturers, third party logistics providers, and distributors to manufacture, store, and move product. Cisco's global supply chain can be separated into two quasi-independent networks: the manufacturing network and the distribution network (1).

The manufacturing network consists of suppliers, and two types of nodes:

- Manufacturing nodes: PCBA and DF
- Warehousing nodes: GHUB

A customer order initiates the pull of components from suppliers. Components then travel through manufacturing nodes, and the final product is fed to the distribution network. A majority of the components and work-in-progress products pass through GHUBs before being processed in their associated manufacturing nodes.

The distribution network consists of customers, gateways, and warehousing nodes (origin and destination SLCs). Final products may move from DFs to customers. However, they are mostly routed to SLCs before reaching the customers.

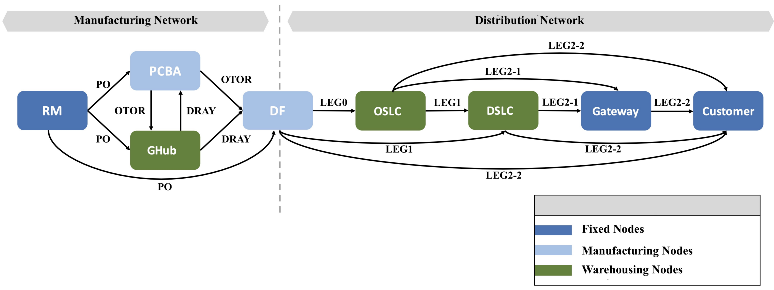

Figure 1: Supply Chain Flow Diagram

SLCs are located in major customer theatres. Each GHUB is located in proximity to a manufacturing node, and only supplies that manufacturing node. These nodes span the Americas, Europe and Asia-Pacific.

Movements between nodes are carried out via Air, Truck, Ocean, or Rail. In reality, Cisco's transportation mode mix varies across time periods.

3 Supply Chain Mathematical Formulation

Networks are mathematical models in which material flows along edges and between nodes. A network is an instance of a graph $\mathcal{G} = (\mathbb{V}, \mathcal{E})$: a finite set of points - vertices/nodes $\mathbb{V}$- and a set $\mathcal{E} \subset \mathbb{V} \times \mathbb{V}$ - edges/arcs. Typically, the data required to describe a network are a supply amount at each node, a cost per unit of flow for each arc, lower and upper bounds on flow along each arc and on the capacities at each node, in addition to the description of the underlying graph. Standard network models always follow two rules of flow conservation:

- **Conservation of flow in an arc**: Flow is neither lost nor gained within an arc.
- **Conservation of flow at nodes**: The supply at a node, plus the total flow into the node, equals the total flow out of the node.

And so, networks are an appropriate model of Cisco's Supply Chain.

3.1 The Network Formulation

In addition to $\mathbb{V}$ and $\mathcal{E}$,

- Let $\mathcal{P}$ be the set of product families
- Let $\mathcal{M}$ be the set of transportation modes

A node in the supply chain may serve multiple product families. Hence, we specify nodes for each product family.

- $\mathbb{V}_{\mathcal{P}} : (\mathbb{V} \times \mathcal{P})$ is the set of nodes defining the supply chain network. Let v_{ip} denote an element of $\mathbb{V}_{\mathcal{P}}$.
- $\mathcal{E}_{\mathcal{P}} : (\mathbb{V}_{\mathcal{P}} \times \mathbb{V}_{\mathcal{P}})$ is the set of directed edges between product specified nodes. Let (v_{ip}, v_{jp}) denote an element of $\mathcal{E}_{\mathcal{P}}$.
- $\mathcal{E}_{\mathcal{M}} : (\mathbb{V} \times \mathbb{V} \times \mathcal{M})$ is the set of directed edges between nodes for each transportation mode. Let (v_i, v_j, m) denote an element of $\mathcal{E}_{\mathcal{M}}$.
- $\mathcal{E}_{\mathcal{PM}} : (\mathbb{V}_{\mathcal{P}} \times \mathbb{V}_{\mathcal{P}} \times \mathcal{M})$ is the set of directed edges between product specified nodes for each transportation mode. Let (v_{ip}, v_{jp}, m) denote an element of $\mathcal{E}_{\mathcal{PM}}$.
- $\mathcal{G}_p : (\mathbb{V}_p,\ \mathcal{E}_p)$ is a product specified graph, defined for all $p \in \mathcal{P}$.

The complete network $\mathcal{G} = \bigcup\limits_{p \in \mathcal{P}} \mathcal{G}_p$. More practically, $\mathcal{G}_p$ comprises all suppliers, manufacturing nodes, warehousing nodes, customers, and associated edges necessary for the production and delivery of product $p \in \mathcal{P}$.

$\mathcal{G}_p$ may be subsequently referred to as a product flow, or pflow. Also, it may be useful to view pflows as the union of paths, $\bigcup\limits_i path_i$, where a path is a sequence of connected and distinct nodes in $\mathcal{G}_p$.

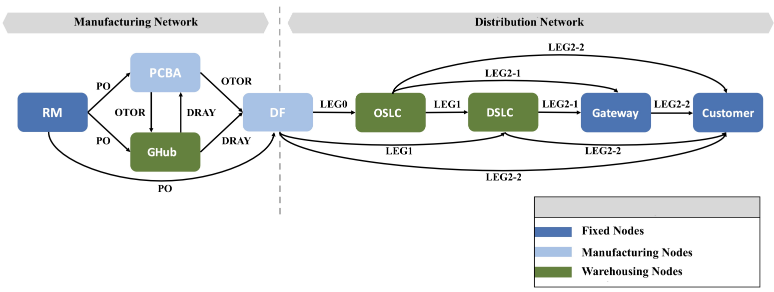

Figure 2: Pflow Diagram

The resulting pflow visualization is familiar and intuitive, convenient attributes of our supply chain model.

3.1.1 Demand

The last component of a sufficient network model is customer demand D, quantified as negative supply S for each product, at each customer node: D_{cp}, where $c \in \mathcal{C}$, the set of customers. $\mathcal{G}_p$ may be further specified in relation to $\mathcal{C}$, $\mathcal{G}_{cp}$, as the pflow of $p \in \mathcal{P}$ for $c \in \mathcal{C}$.

3.2 Scenarios

The set of scenarios is infinite; an observation that may initially seem overwhelming, yet indicates a wide range of model applications. A scenario is a modification of a given network model. All modifications are the result of three "actions": add a node, remove a node, change customer demand. More formally, actions are functions on the sets $\mathbb{V}$ and $\mathcal{E}$, or on D.

Another useful concept is the Mother Scenario and is dynamically defined as $\mathcal{G}_{mother}$ s.t. all scenario networks are a subset of $\mathcal{G}_{mother}$.

3.2.1 Add a Node

Let f_+ be the "add a node" function. In practice, adding a node entails adding a supplier, warehouse, or manufacturing facility to the supply chain. The node would serve a subset of $\mathcal{P}$, and all of $\mathcal{C}$. In the context of our model, f_+ creates alternate pflows: given a new node v^1, which serves the same role as v, $f_+(v^1, \textit{pflow})$ returns *pflow'*, such that *pflow* differs from *pflow'* in the subset of its constituting paths that contain v. Moreover, f_+ returns the set of edges necessary to create the new differing paths.

For example, let *pflow* = $\textit{pth}_1 \cup \textit{pth}_2$, s.t. $\textit{pth}_1 = \{v_1, v_3, v_4\}$ and $\textit{pth}_2 = \{v_2, v_3, v_4\}$. Given a new node v_1^1, that serves the same role as v_1: $f_+(v_1^1, \textit{pflow}) = \{v_1^1, v_3, v_4\} \cup \{v_2, v_3, v_4\}$ and $\{v_1^1, v_3, m\} \forall m \in \mathcal{M}$.

3.2.2 Remove a Node

Let f_- be the "remove a node" function. In practice, removing a node entails removing a supplier, warehouse, or manufacturing facility from the supply chain. In the context of our model, f_- creates alternate pflows: given a node v, $f_-(v, \textit{pflow})$ returns *pflow'*, such that *pflow* differs from *pflow'* in the subset of its constituting paths that contain v. Moreover, f_- removes the set of edges necessary to create the new differing paths.

For example, let *pflow* = $\textit{pth}_1 \cup \textit{pth}_2$, s.t. $\textit{pth}_1 = \{v_1, v_3, v_4\}$ and $\textit{pth}_2 = \{v_2, v_3, v_4\}$. Given v_1: $f_-(v_1, \textit{pflow}) = \{Nan, v_3, v_4\} \cup \{v_2, v_3, v_4\}$ and $-\{v_1, v_3, m\}\ \forall m \in \mathcal{M}$.

3.2.3 Change Customer Demand

Let f_d be the "change customer demand" function. The function returns a matrix $D^{|\mathcal{C}| \times |\mathcal{P}|}$, s.t. D_{cp} = new demand of customer $c \in \mathcal{C}$ for product family $p \in \mathcal{P}$.

4 Data to Model

The mathematical framework is now defined, and ready to be populated with data: arcs and nodes must be associated with qualitative and quantitative characteristics.

4.1 Available Data

Lane Rate Data is the major data source available and adequate to populate the network model. The data is an edge dictionary, a record of all product movements between nodes for a certain period of time. $\mathcal{Z}$ denotes the edge dictionary.

In addition, a multitude of term dictionaries define the qualitative characteristics of edge attributes in the edge dictionary.

4.2 Node Data

In *Supply Chain Overview*, node roles are defined as per their operational function. For example, a supplier (RM) is the source of all components and materials used in the manufacturing of Cisco products. A more amenable description would do so in terms of elements in the model, i.e. edges (shipment_type).

Let $RL = \{$*Error, Supplier/RM, GHUB, PCBA, DF, OSLC, DSLC, Gateway/GWY, Customer*$\}$ be the set of node roles, $TP = \{$*PO, OTOR, DRAY, LEG0, LEG1, LEG2-1, LEG2-2*$\}$ be the set of edge/shipment types, and $f_{nd} : IN \times OUT \text{ –> } RL$ be an alternative node role definition, where:

- $IN \subset TP$, the set of inbound edge types to a node
- $OUT \subset TP$, the set of outbound edge types from a node

Indeed, f_{nd} is onto, and maps:

- $\emptyset \times \{PO\} \longrightarrow RM$
- $\{PO, OTOR\} \times \{DRAY\} \longrightarrow GHUB$
- $\{PO, DRAY\} \times \{OTOR\} \longrightarrow PCBA$
- $\{PO, OTOR, DRAY\} \times \{LEG0, LEG1, LEG2\text{-}1\} \longrightarrow DF$
- $\{LEG0\} \times \{LEG1, LEG2\text{-}1, LEG2\text{-}2\} \longrightarrow OSLC$
- $\{LEG1\} \times \{LEG2\text{-}1, LEG2\text{-}2\} \longrightarrow DSLC$
- $\{LEG2\text{-}1\} \times \{LEG2\text{-}2\} \longrightarrow GWY$
- $\{LEG2\text{-}2\} \times \emptyset \longrightarrow Customer$
- $TP \times TP - \{\textit{afore-defined domains}\} \longrightarrow Error$

Furthermore, let us introduce two methods to order nodes. First, a partial order on shipment types, hence developing a hierarchy of edges:

Shipment Type	**Order**
PO	0
OTOR1	1
DRAY	1
LEG0	2
LEG1	3
LEG2-1	4
LEG2-2	5

Table 1: Partial Order of Shipment Type

Second, a node ordering in relation to paths. Let $pth = \{v_1, v_2, \ \dots \ , v_{n-1}, v_n\}$ be a path in a pflow. The rank of v_i = the number of edges separating v_i from the final node in the path. Note: The final node in a path is always a customer.

All the above notation and concepts surrounding nodes may seem unnecessary, yet will prove to be useful concepts in developing the Scenario Modeling Tool, informing database design and algorithms.

In addition, nodes have the following quantitative characteristics:

- Capacity, an upper bound on material flowing into a node: $U^{1\times|\mathbb{V}|}$ in kg
- Variable Operations Cost (OPEX): $V^{1\times|\mathbb{V}_{\mathcal{P}}|}$ in \$/kg
- Fixed Operations Cost (CAPEX): $R^{1\times|\mathbb{V}_{\mathcal{P}}|}$ in \$
- Demand/Supply: $D^{1\times|\mathbb{V}_{\mathcal{P}}|}$ in kg

Node Type	**Node Characteristics**
Supplier	D
PCBA	U, V, R, D
DF	U, V, R, D
GHUB	U, V, R, D
xSLC	U, V, R, D
Gateway	U, V, R, D
Customer	D

Table 2: Breakdown of Node Characteristics

4.3 Edge Data

The main data source is an edge dictionary, conveniently setting the edges in our system. Edges have the following quantitative characteristics:

- Transportation Cost: $C^{1\times|\mathcal{E}_{\mathcal{PM}}|}$
- Transportation Time: $T^{1\times|\mathcal{E}_{\mathcal{PM}}|}$
- Carbon Emissions: $E^{1\times|\mathcal{E}_{\mathcal{PM}}|}$

4.4 Demand Data

Demand, $D_{\mathcal{CP}}$ may be generated from historical data, or forecasting.

4.5 New Scenario Data

Data needs for generating a new scenario are best evaluated according to the three core actions: "Add a Node - f_+", "Remove a Node - f_-", "Change Customer Demand - f_d".

f_- does not require new data, and f_d is self-evident. Adding a node v_i^1 similar to v_i, is equivalent to $f_+(v_i^1, l)\forall l \in G$. Given a path $h = \{v_1, \dots, v_i, \dots, v_n\} \in l$ s.t. $v_i \in h$, we have $h^1 = \{v_1, \dots, v_i^1, \dots, v_n\}$. And so, $\forall l \in G, v_i \in l$:

$$f_+(v_i^1, l) = f_+(v_i,\ (\bigcup_{h\in l, v_i\in l} h) \cup (\bigcup_{h\in l, v_i\notin h} h)) = l^1 \text{ \& necessary edges} \tag{1}$$

$$l^1 = ((\bigcup_{h^1\in l^1, v_i\in l^1} h^1) \cup (\bigcup_{h\in l, v_i\notin h} h)) \ \ \& \ \ \{(v_{i-1}, v_i^1, m),\ (v_i^1, v_{i+1}, m) \mid \forall m \in \mathcal{M}\} \tag{2}$$

Consequently, estimates for $U_{v_i^1}$, $B_{v_i^1}$, $R_{v_i^1}$, $D_{v_i^1}$, as well as estimates for $C_{v_{i-1},v_i^1,m}$, $T_{v_{i-1},v_i^1,m}$, $E_{v_{i-1},v_i^1,m}$, $C_{v_i^1,v_{i+1},m}$, $T_{v_i^1,v_{i+1},m}$, $E_{v_i^1,v_{i+1},m}$ $\forall m \in \mathcal{M}$ are necessary to conduct scenario modeling.

4.6 Measures of Risk

Supply Chain Risk is an unavoidable factor in determining network design, and is usually discussed in conjunction and disjunction to outputs of similar models, as risk tends to escape the reach of common mathematical formulations.

Nevertheless, there are many a way to capture elements of risk in network models, notably:

- Disturbing data deemed to be most affected by elements of risk, or risk collateral; i.e. different costs.
- Applying penalties to edges and nodes deemed to be close to sources of risk; i.e. flow through nodes and on edges in areas of high geopolitical risk.

Both approaches are equivalent to applying penalties to risk collateral, and/or sources of risk. They require the model to be further encumbered with a multitude of assumptions, namely, what constitutes risk collateral and sources of risk. Also, a lengthy penalty tuning process is necessary to ensure the penalties don't unduly jeopardize flow integrity.

A third approach entails Leech Nodes, which are introduced below.

4.6.1 Leech Nodes

Let $\mathbb{V}_p^L$ be the set of leech nodes, $v_{ip}^L \in \mathbb{V}_p^L$ be a leech to v_{ip}, $(v_{ip}, v_{ip}^L) \in \mathcal{E}_p$ be a leech edge, and $D_{v_{ip}}^L$ be the "leech demand". Leech Nodes latch on to manufacturing nodes, and leech a specific amount of product family $p \in \mathcal{P}$ from those nodes. Consequently, leeches tax the pflows $\mathcal{G}_{cp}$ s.t. $v_{ip} \in \mathcal{G}_{cp}$, as more flow is required to fulfill customer demand. It is reasonable to state that the higher the measure of risk associated with a node or collection of nodes, the higher the leech demand.

Moreover, Risk is a random process, best modeled as a random variable or a stochastic process. As such, individual $D^L_{v_{ip}}$ may be modeled as RVs, or as collections of RVs (a stochastic process): $D^L_{v_{ip}}\ \forall\ v_{ip}$ along a path.

4.7 Limitations & Proposed Mitigation

The astute reader may have caught two glaring omissions. There is no mention of manufacturing processes, which is paramount in determining the amount of flow on arcs. Also, there is no mention of data fallacies and their effect on building the network model.

4.7.1 Manufacturing Processes

Strategic decisions inform whether components go directly to manufacturing nodes, or journey through warehousing nodes. Also, the network must ensure manufacturing nodes receive all necessary components in adequate quantities to successfully contribute to the pflow. Sadly, acquiring data sources explicitly relaying such information is unwieldy ... For instance, there is no central repository of bill of materials $\forall pin\mathcal{P}$.

However, the edge dictionary (1) must reflect the truth (!): strategic routing decisions, and the component mix ensuring proper manufacturing node output. Namely, product was successfully manufactured and delivered to customers, implying that manufacturing nodes have received adequate quantities of components from adjacent nodes to output product.

$\forall v_{ip} \in \mathbb{V}_{\mathcal{P}},\ \forall v_{jp} \in \mathbb{V}_{\mathcal{P}}$ s.t. $(v_{ip}, v_{jp}) \in \mathcal{E}_{\mathcal{P}}$, let

$$\alpha_{ijp} = \frac{\sum\limits_{(i,j,p)\in\mathcal{Z}} \text{BILLED\_WEIGHT}}{\sum\limits_{i} \sum\limits_{(i,j,p)\in\mathcal{Z}} \text{BILLED\_WEIGHT}} \tag{3}$$

α is sufficient to enforce flow integrity in the network; consider that there exists a recipe $\alpha\ \forall\ p \in \mathcal{P}$, where adjacent nodes i are the ingredients. α_{ijp} is the proportion of ingredient i for one unit of output p from node j.

4.7.2 Node Capacity

Beyond product recipes, there is no central data set on node capacity. In the context of the α metaphor, node capacity is the maximum amount of ingredients a node j can process and output.

Given a path $pth = \{v_{1p}, v_{2p}, \ \ldots\ , v_{n-1p}, v_{np}\}$ and α, we can calculate the proportion of finished product flow ($total\_\alpha_{ip}$) at each node $v_i \in pth$:

$$total\_\alpha_{ip} = \Pi_{j=i}^{n-1} \alpha_{jj+1p} \tag{4}$$

There exists readily available data on historical node usage and total demand $D_p\ \forall p \in \mathcal{P}$. Thus,

$$\text{U}_i \approx \frac{\sum_{p\in\mathcal{P}} D_p \times total\_\alpha_{ip}}{\text{usage of } v_i} \tag{5}$$

4.7.3 Data Inconsistencies

Data can be corrupted in three ways: misspellings, mislabeling, omissions. Reconciling different spellings of node instances is pedantic, yet routine. Mislabeling may be remedied by enforcing the aforementioned (4.2) node rankings on the data. And most omissions may be deciphered in relation to adjacent edges in $\mathcal{Z}$. Seldom, corrupted data may resist all remedies. α conveniently curtails the subsequent corruption of the model, as its assigned weight or importance within pflows will be minimal.

$\mathcal{Z}$ is corrupted such that the role of nodes is not evident in row attributes, and has to be inferred. The Network Reconstruction Protocol (NRP) reveals the structure of the global supply chain and identifies node roles in $\mathcal{Z}$.

4.7.3.1 Network Reconstruction Protocol

The main insight driving the NRP follows from the node notation and concepts introduced in Node Data: given a path $pth = \{v_1, v_2, \ldots, v_{n-1}, v_n\}$ and the alternative node definition f_{nd}, one could determine the role of $v_i \in pth$. Thus, the NRP comprise two algorithms:

- The Path Finder Algorithm (PFA)
- The Node Role Algorithm (NRA)

The PFA finds and groups all paths in $\mathcal{Z}$, revealing the structure of the network $\mathcal{G}$. The NRA applies f_{nd} to all paths $\in \mathcal{G}$, determining the roles of nodes in the network. Hence, the NRP outputs an accurate representation of the state of the supply chain during the period of $\mathcal{Z}$.

5 The Network Optimization Model

5.1 Why Network Optimization?

Network Models are very easy to depict visually, and so are easy to "sell" to the layman. Furthermore, they can be solved significantly more efficiently than LPs, enabling the model to handle approximately 10 to 100 times as many variables. Basic Feasible Solutions to a network problem are guaranteed to be integer, as long as supplies and bounds are integer. Therefore, solving network problems is equivalent to solving integer programs, which model the yes/no (binary) decisions prevalent in real problems.

5.2 Typical Structure

The typical network problem aims to fulfill supply and demand requirements at minimum total cost, without violating bounds on arc flows. Given a network $G = (V, E)$, with data $s_j \geq 0$ denoting the supply at node $j \in V$ and $s_j \leq 0$ denoting the demand at node $j \in V$ s.t. $\sum_{j \in V} s_j = 0$. The variables are $x_{ij} \in E$, with associated costs c_{ij}, lower bounds l_{ij}, and upper bounds u_{ij}. The LP for this problem follows.

$$\min \sum_{(i,j) \in E} c_{ij} x_{ij} \quad \text{s.t.} \tag{6}$$

$$\sum_j x_{ij} - \sum_j x_{ji} = s_i \quad \forall i \in V \tag{7}$$

$$l_{ij} \leq x_{ij} \leq u_{ij} \quad \forall (i,j) \in E \tag{8}$$

5.3 Assumptions

The model' assumptions:

- Weight remains constant throughout product transformation
- There is no seasonality to customer demand
- The manufacturing and delivery schedules are ideal and perfectly respected

5.4 The Optimization Model - MILP

5.4.1 Indices

$n, i, j, k = 1...N$ Nodes;

$p = 1...P$ Products;

$m = 1...M$ Transportation modes;

$a = \{$cost, carbon, time$\}$;

5.4.2 Intra & Inter Index Relationships

IN$(j, p) = i \quad$ s.t. $(i, j, p) \in \mathcal{E}_{\mathcal{P}}$, returns the set of inbound nodes to j.

OUT$(j, p) = k \quad$ s.t. $(j, k, p) \in \mathcal{E}_{\mathcal{P}}$, returns the set of outbound nodes from j.

TM$(i, j) = m \quad$ s.t. $(i, j, m) \in \mathcal{E}_{\mathcal{M}}$, returns the set of possible transportation modes on the edge (i, j).

5.4.3 Decision Variables

X_{ijpm} = Flow of product p leaving node i to node j by mode m (in kg).

5.4.4 Auxiliary Variables

O_n = 1 if node n is operational, 0 otherwise.

5.4.5 Input Data

C_{ijpm} = Transportation cost for flow of product p leaving node i to node j by mode m (in \$/kg).

V_{np} = OPEX at node n for product p (in \$/kg).

R_n = CAPEX for node n (in \$).

T_{ijm} = Transportation time from node i to node j by mode m (in days).

E_{ijm} = Carbon emissions from node i to node j by mode m (in CO2e/kg)

U_n = Capacity of node n (in kg).

S_{np} = Supply/Demand for node n for product p.

α_{ijp} = Percentage of flow from node i to node j used in the transformation of product p at node i. (4.7.1)

λ_a = Weight for objective function a.

ω_a = Inverse baseline value for a. As the objectives are represented in different units at differing magnitudes, they are divided by the actual value of a for the current network. The resulting objective function is a weighted linear sum of the three objectives. By varying these weights, the model derives a series of Pareto efficient solutions that range in performance across the objectives.

5.4.6 Objective

$$\begin{aligned}\text{Minimize}\quad & \lambda_{\text{cost}}\left(\sum_{j}\sum_{i\in\text{IN}(j,p)}\sum_{p}\sum_{m\in\text{TM}(i,j)} X_{ijpm}(C_{ijpm}+V_{ip})+\sum_{n} O_nR_n\right)\times\omega_{\text{cost}}\\ & +\lambda_{\text{carbon}}\left(\sum_{j}\sum_{i\in\text{IN}(j,p)}\sum_{p}\sum_{m\in\text{TM}(i,j)} X_{ijpm}E_{ijm}\right)\times\omega_{\text{carbon}}\\ & +\lambda_{\text{time}}\left(\sum_{j}\sum_{i\in\text{IN}(j,p)}\sum_{p}\sum_{m\in\text{TM}(i,j)} X_{ijpm}T_{ijm}\right)\times\omega_{\text{time}}\end{aligned}\tag{9}$$

5.4.7 Constraints

$$\sum_{k\in\text{OUT}(j,p)}\sum_{m\in\text{TM}(j,k)} X_{jkpm}-\sum_{i\in\text{IN}(j,p)}\sum_{m\in\text{TM}(i,j)} X_{ijpm}=S_{jp}\quad\forall j,p\tag{10}$$

$$\sum_{i\in\text{IN}(j,p)}\sum_{m\in\text{TM}(i,j)}(X_{ijpm})\le U_jQ_j\quad\forall j,\ p\tag{11}$$

$$\sum_{m\in\text{TM}(i,j)} X_{ijpm}=\alpha_{ijp}\left[\sum_{k\in\text{OUT}(j,p)}\sum_{m\in\text{TM}(j,k)} X_{jkpm}-S_{jp}\right]\quad\forall i,j,p\tag{12}$$

$$X_{ijpm}\ge 0\quad\forall i,j,p,m\tag{13}$$

$$O_n\in\{0,1\}\quad\forall n\tag{14}$$

Constraint (10) ensures conservation of flow in arcs and at nodes.

Constraint (11) ensures that node capacities cannot be exceeded.

Constraint (12) ensures manufacturing processes are respected.

Constraints (13) and (14) specify the domain of the variables.

5.4.8 Different Scenarios

As detailed in Scenarios, a new scenario is the result of three "actions": add a node, remove a node, change customer demand. While changing customer demand is merely a change in input data, the other two actions require alterations to the MILP; alterations and additions of constraints.

5.4.8.1 Adding Nodes

Suppose the scenario aims to add nodes $\{v^1_{ip},\dots,v^a_{ip}\}$, serving the same role as v_{ip} in $\mathcal{G}_p$. Evidently, all nodes in $\{v_{ip},v^1_{ip},\dots,v^a_{ip}\}$ are capable of fulfilling the role of v_{ip}. For ease of notation, let v_{ip} be v^0_{ip}, the original node. Also, in MILP notation, $\{i^0,i^1,\dots,i^a\}\equiv\{v^0_{ip},v^1_{ip},\dots,v^a_{ip}\}$.

Flow conservation at and around those nodes must be observed, their capacities should not be exceeded, and the associated decision and auxiliary variables should have the same domains. And so, constraints 10, 11, 13 and 14 remain unchanged. However, constraint 12 should be modified to maintain flow integrity in $\mathcal{G}_p$, i.e. respect manufacturing processes for $p\in\mathcal{P}$:

$$\sum_{b=0}^{a}\sum_{m\in\text{TM}(i^a,j)} X_{i^ajpm}=\alpha_{ijp}\left[\sum_{k\in\text{OUT}(j,p)}\sum_{m\in\text{TM}(j,k)} X_{jkpm}-S_{jp}\right]\quad\forall i,j,p\tag{15}$$

5.4.8.2 Removing Nodes

Simply removing a node would result in a disconnected network, and an infeasible model. Consequently, in the definition of new scenarios, f_- has to be accompanied with f_+, essentially replacing a node v_{ip} in $\mathcal{G}_p$ with nodes $\{v_{ip}^1, \ldots, v_{ip}^a\}$; or $\{i^1, \ldots, i^a\}$ in MILP notation.

As in adding a node, flow conservation at and around those nodes must be observed, their capacities should not be exceeded, and the associated decision and auxiliary variables should have the same domains. And so, constraints 10, 11, 13 and 14 remain unchanged. However, constraint 12 should be modified to maintain flow integrity in $\mathcal{G}_p$, i.e. respect manufacturing processes for $p \in \mathcal{P}$:

$$\sum_{b=1}^{a} \sum_{m \in \text{TM}(i^a, j)} X_{i^a jpm} = \alpha_{ijp} \left[\sum_{k \in \text{OUT}(j,p)} \sum_{m \in \text{TM}(j,k)} X_{jkpm} - S_{jp} \right] \quad \forall i, j, p \tag{16}$$

Furthermore, the MILP must require i^0 be "off", equivalent to setting the auxiliary variable

$$O_{i^0} = 0 \tag{17}$$

5.4.8.3 Limiting Alternative Node Count

Suppose the scenario aims to operate a subset of the set of alternatives $\{i^0, i^1, \ldots, i^a\}$, s.t. a number $a' \leq a$ of those nodes are operational. The MILP must enforce the following constraint

$$\sum_{a=0}^{n} O_{i^a} = a' \tag{18}$$

Also, if the scenario requires some subset I of the set of alternatives to be operational:

$$O_n = 1 \quad \forall n \in I \tag{19}$$

6 Model Design

A data pipeline transforms, melds, and funnels a multitude of data sources into the MILP, revealing ideal and optimal states of the global supply chain. Moreover, output derivatives present results in more palatable formats, best suited for the stakeholders and intended end-users.

Scenario modeling requires batches of MILP runs, as a multitude of runs are necessary to capture the impact of varying the weight of our objectives and the impact of risk on the optimal solution.