15.089: ANALYTICS CAPSTONE
MINIMIZING VACCINE VARIANCE

Kyle Mana and Ryan Trusler
Faculty Advisor: Daniel Freund

AGENDA

01 Background & Scope
- *Project Scope and Aims*

02 Modeling Supply Chain
- *Digitizing supply chain flows*

03 Optimization
- *Formulation and impact*

04 User Interface
- *Platform and interface*
- *Design*

05 Impact
- *Cost savings*
- *Time savings*

BRAND BACKGROUND

- **Prevenar Brand**
  - *Pneumococcal conjugate vaccine*
  - *Used across infants, children and adults*
  - *Given as a four-dose vaccine for children under 6 and as a single does otherwise*
  - *Regionally “Prevnar” in North America*
- **Opportunity within Prevenar**
  - *Global brand with presence in markets in 6 continents*
  - *With a partnership with UNICEF, the brand is used in virtually every country worldwide*
  - *World’s most used vaccine before COVID*
  - *$5.95 billion in sales in 2020*

Prevnar 20

Prevenar 13®

Pneumococcal polysaccharide conjugate vaccine, 13-valent adsorbed

OUR SCOPE

Currently:

Brand managers use excel-based tools to evaluate brand health and long-term plans

- These tools are fragile, memory intensive, and rigid
- Re-evaluating inventory plans for different scenarios and strategies is a time-intensive exercise
- Each tool requires manual data extracts from 2+ data sources
- Heuristics such as “round-up” ordering policies are used to plan inventory

Our objectives are to:

Offer a robust, stable, and fit-for-purpose interface for brand reporting and planning

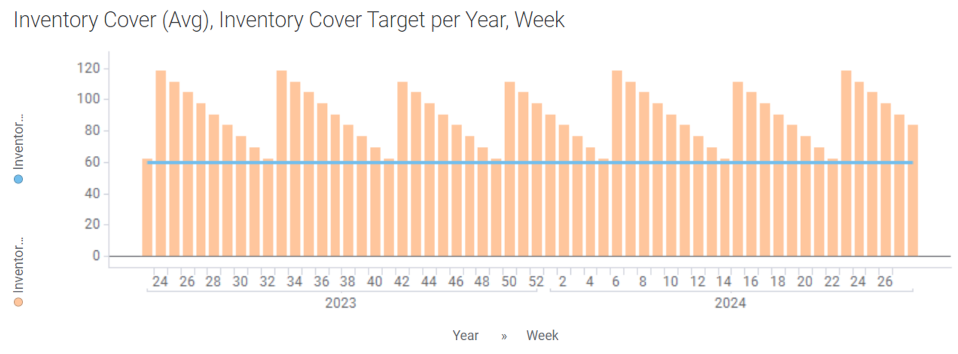
- Develop a tool that can generalize to any brand, and any network
- Allow for quick “what-if” scenario planning across multiple forecasts
- Reduce the overhead required to perform these analysis using automation
- Thoughtfully insert optimization in the place of heuristics

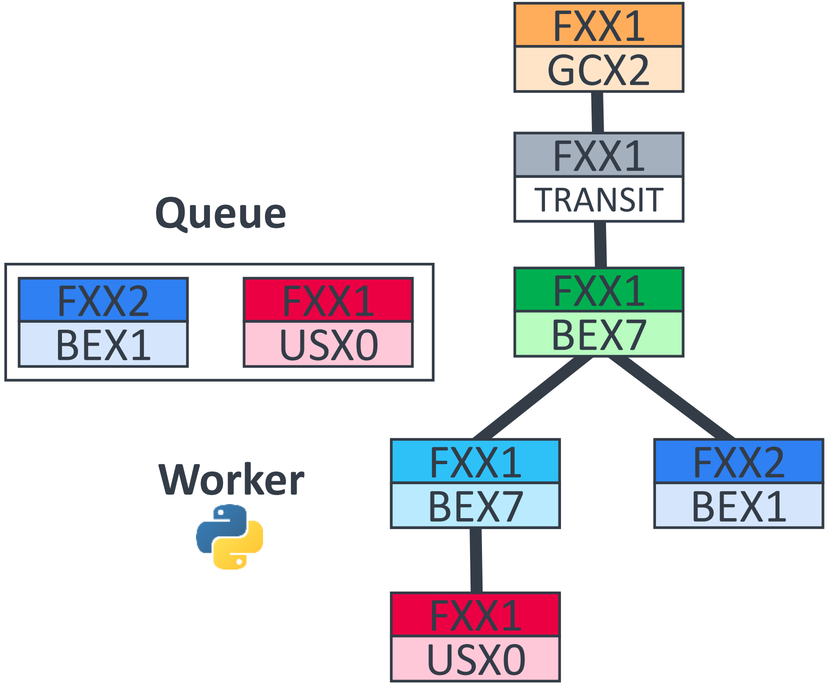
REPLICATING SUPPLY CHAIN LOGIC

Graph representation of supply chain

- Nodes represented as product/location pairs
- Demand is pulled down the supply chain by the terminal nodes (leaves)
- Genealogy used to create a "digital twin" of the supply chain

Round-up Strategy:





OPTIMIZATION - FORMULATION

Minimize a weighted combination of inventory and deviation from target inventory

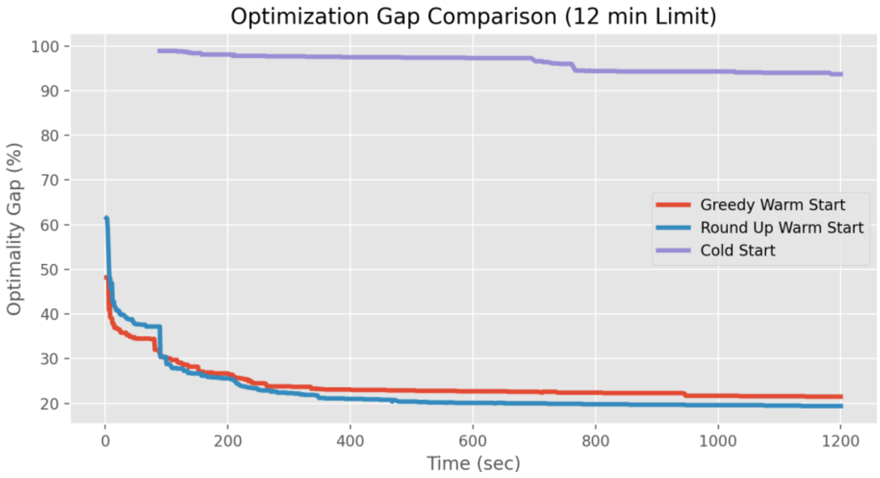
Subject to:

- ***Flow Constraint:*** *Inventory at a node must equal starting inventory plus flows in, minus flows out*
- ***Satisfy Demand:*** *Always satisfy demand at the terminal nodes*
- ***Inventory Threshold:*** *Never drop below a specified percentage of the target inventory*
- ***Target Inventory:*** *Equal to the inventory required to satisfy demand over the desired cycle time (considering lead times, manufacturing times and safety stock)*

OPTIMIZATION – WARM START

- **Solving the optimization un-aided was deemed intractable (12 min run-time limits)**
- **We implemented two warm-start heuristics to provide feasible integer solutions**
  1. *The current round-up method*
  2. *A greedy optimization, which makes the best decision at a single node moving up the graph*
- **Benefits: faster optimization by giving the solver a strong target to "do better than"**

Time	Cold Start Gap %	Round-up Warm Start Gap %	Greedy Warm Start Gap %
0		61.5	48.2
100	98.9	28.8	30.1
200	98.1	25.6	26.7
300	97.7	22.3	23.8
400	97.5	21	23.1
500	97.4	20.4	22.9
600	97.3	20.1	22.8
700	96.6	20	22.6
800	94.4	19.8	22.4
900	94.3	19.7	22.3
1000	94.3	19.6	21.7
1100	94	19.5	21.6
1200	93.7	19.4	21.5



OPTIMIZATION – RESULTS & IMPACT

Our weekly optimization identifies opportunity to reduce planned inventory by

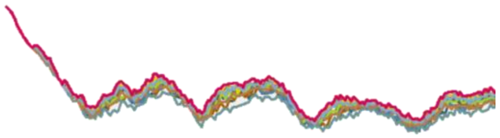
↓11%

which translates to annual savings of

$4M

for the Prevenar brand

We enable frictionless evaluation of inventory health across several inventory strategies



Brand managers get time back, all while planning to a higher level of detail

Before	After
Several manual data pulls per analysis	Automated data pipeline
Monthly granularity	Weekly or monthly granularity
Rigid round-up ordering policy	Round-up, greedy, and optimized policies
Repeat entire process per scenario	Compare hundreds of scenarios at once

MIT MANAGEMENT SLOAN SCHOOL

THANK YOU

MIT Sloan School of Management
50 Memorial Drive, E52-359 Cambridge, MA 02142 USA
617-258-5434 sfadmissions.mitsloan@mit.edu

#MyMITSloan **mitsloan.mit.edu**

APPENDIX

OPTIMIZATION - FORMULATION

$$\min_{\alpha,\sigma,\omega} \quad \sum_{d,n} \alpha_{dn}(1-\lambda)\theta + \sigma_{dn}\lambda$$

$$\text{s.t.} \quad s_n + \sum_{t=0}^{d} \sum_{j:(j,n)\in\mathcal{A}} \omega_{tjn} u_n - \sum_{t=0}^{d} \sum_{j:(n,j)\in\mathcal{A}} \omega_{tnj} u_j = \sigma_{dn} \quad \forall d \in D, \forall n \in N \setminus Sink \tag{2}$$

$$\sum_{j:(j,n)\in\mathcal{A}} \omega_{djn} u_n = f_{dn} \qquad d \in D, \forall n \in Sink$$

$$\sigma_{dn} \geq K_{dn} e_n \qquad \forall d \in D, \forall n \in N \tag{3}$$

$$\alpha_{dn} \geq K_{dn} - \sigma_{dn} \qquad \forall d \in D, n \in N \tag{4}$$

$$\alpha_{dn} \geq -(K_{dn} - \sigma_{dn}) \qquad \forall d \in D, n \in N \tag{5}$$

$$K_{dn} = \sum_{j:(n,j)\in\mathcal{A}} \sum_{t=d+1}^{d+c_{dn}} \omega_{tnj} u_j \qquad \forall d \in D, n \in N \setminus Sink \tag{6}$$

$$\omega_{djn} \in \mathbb{Z}_+, \alpha_{dn} \in \mathbb{R}_+, \sigma_{dn} \in \mathbb{R}_+$$

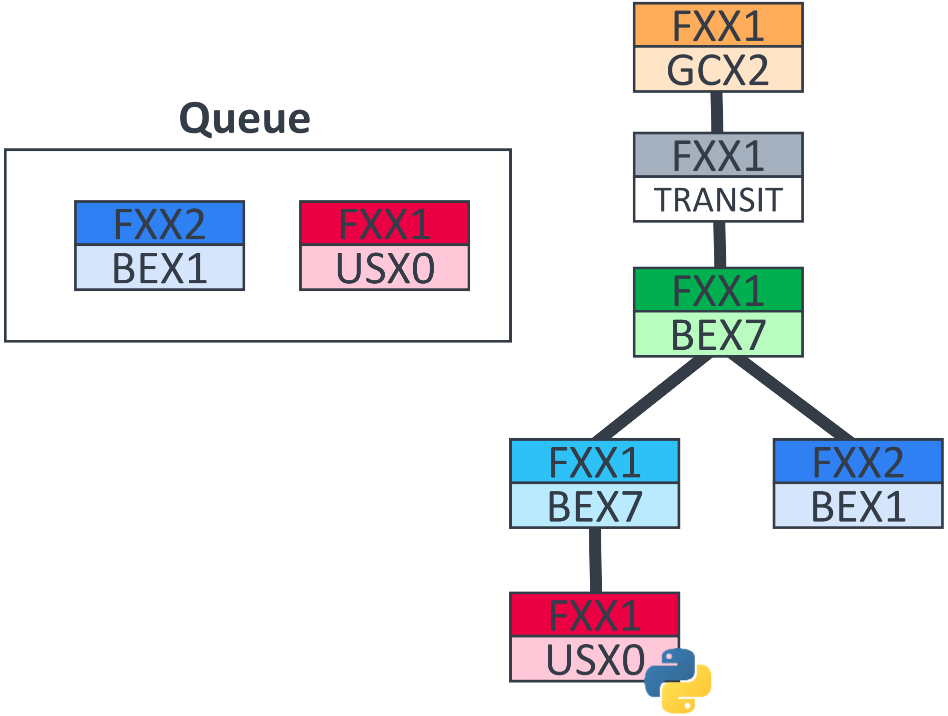
REPLICATING SUPPLY CHAIN LOGIC

Graph representation of supply chain

- Nodes represented as product/location pairs
- Demand is "pulled" down the supply chain by the terminal nodes (leaves)

Tree search algorithm

- Initialize a queue od nodes to visit with terminal nodes
- Add a node to the queue only after all its children have been visited



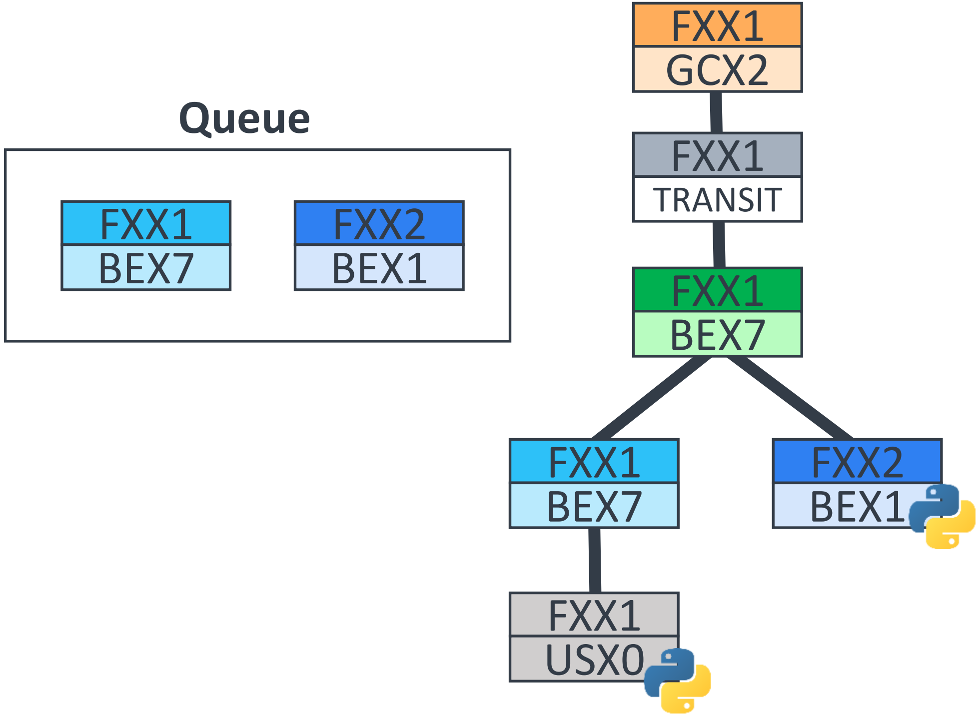
REPLICATING SUPPLY CHAIN LOGIC

Graph representation of supply chain

- Nodes represented as product/location pairs
- Demand is "pulled" down the supply chain by the terminal nodes (leaves)

Tree search algorithm

- Initialize a queue od nodes to visit with terminal nodes
- Add a node to the queue only after all its children have been visited



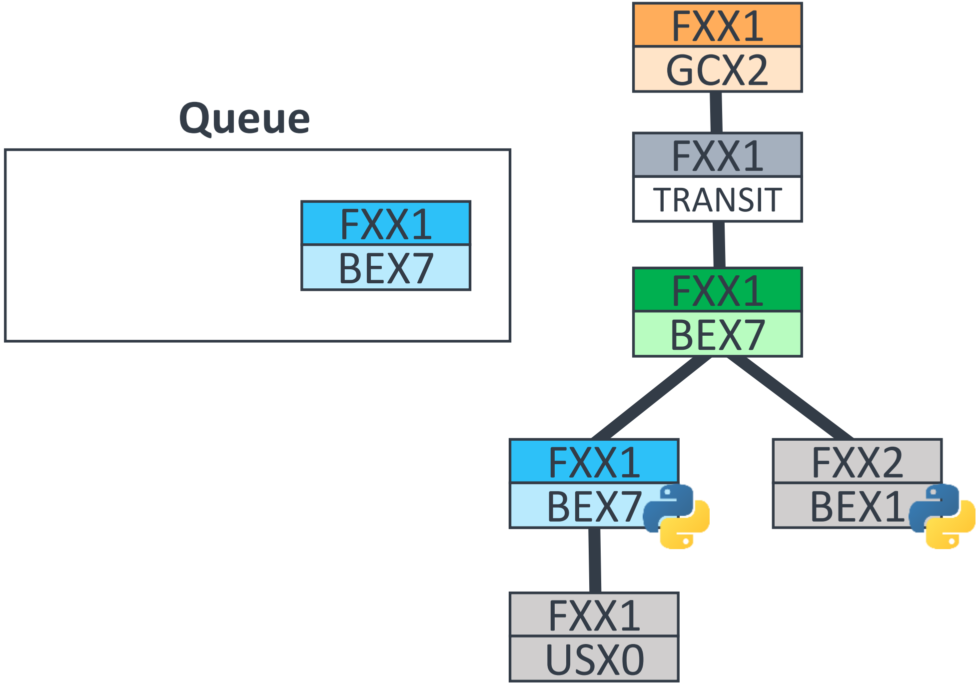
REPLICATING SUPPLY CHAIN LOGIC

Graph representation of supply chain

- Nodes represented as product/location pairs
- Demand is "pulled" down the supply chain by the terminal nodes (leaves)

Tree search algorithm

- Initialize a queue od nodes to visit with terminal nodes
- Add a node to the queue only after all its children have been visited



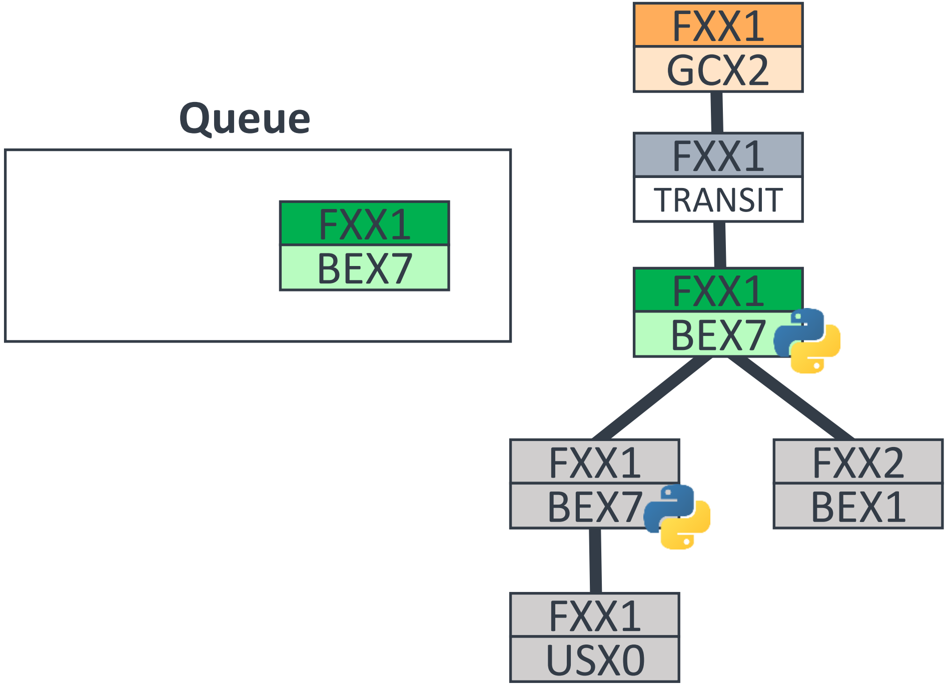
REPLICATING SUPPLY CHAIN LOGIC

Graph representation of supply chain

- Nodes represented as product/location pairs
- Demand is "pulled" down the supply chain by the terminal nodes (leaves)

Tree search algorithm

- Initialize a queue od nodes to visit with terminal nodes
- Add a node to the queue only after all its children have been visited



CAPITAL COST SAVINGS

- **With fixed inventory threshold of 70% and risk tolerance of 0.5**