15.089: ANALYTICS CAPSTONE
MINIMIZING VACCINE VARIANCE

Kyle Mana and Ryan Trusler
Faculty Advisor: Daniel Freund
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
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:

Inventory Cover (Avg), Inventory Cover Target per Year, Week

![Graph example](chart.png)
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)
• 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”

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

<table>
<thead>
<tr>
<th>Before</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>Several manual data pulls per analysis</td>
<td>Automated data pipeline</td>
</tr>
<tr>
<td>Monthly granularity</td>
<td>Weekly or monthly granularity</td>
</tr>
<tr>
<td>Rigid round-up ordering policy</td>
<td>Round-up, greedy, and optimized policies</td>
</tr>
<tr>
<td>Repeat entire process per scenario</td>
<td>Compare hundreds of scenarios at once</td>
</tr>
</tbody>
</table>
APPENDIX
\[
\min_{\alpha, \sigma, \omega} \sum_{d,n} \alpha_{dn}(1 - \lambda)\theta + \sigma_{dn}\lambda \\
\text{s.t.} \quad s_n + \sum_{t=0}^{d} \sum_{j:(j,n)\in A} \omega_{ijn}u_n - \sum_{t=0}^{d} \sum_{j:(n,j)\in A} \omega_{tnj}u_j = \sigma_{dn} \quad \forall d \in D, \forall n \in N \setminus \text{Sink} \\
\sum_{j:(j,n)\in A} \omega_{ijn}u_n = f_{dn} \\
\sigma_{dn} \geq K_{dn}c_n \\
\alpha_{dn} \geq K_{dn} - \sigma_{dn} \\
\alpha_{dn} \geq -(K_{dn} - \sigma_{dn}) \\
K_{dn} = \sum_{j:(n,j)\in A} \sum_{t=d+1}^{d+c_{dn}} \omega_{tnj}u_j \\
\omega_{ijn} \in \mathbb{Z}^+, \alpha_{dn} \in \mathbb{R}^+, \sigma_{dn} \in \mathbb{R}^+ \\
d \in D, \forall n \in \text{Sink} \\
\forall d \in D, \forall n \in N \\
\forall d \in D, n \in N \\
\forall d \in D, n \in N \setminus \text{Sink} \\
(2) \\
(3) \\
(4) \\
(5) \\
(6)
\]
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 of nodes to visit with terminal nodes
- Add a node to the queue only after all its children have been visited
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 of nodes to visit with terminal nodes
- Add a node to the queue only after all its children have been visited
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 of nodes to visit with terminal nodes
• Add a node to the queue only after all its children have been visited
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 of 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

Working Capital Cost Savings ($ Millions)

- Greedy Heuristic Monthly: 3.3525
- Monthly Optimization: 4.5545
- Round Up Weekly Plan: 4.5456
- Greedy Heuristic Weekly: 5.0473
- Weekly Optimization: 5.5531