Inventory optimization for Industrial-Scale assemble-to-order Systems

Willem van Jaarsveld
University Rotterdam - Erasmus School of Economics
&
Alan Scheller-Wolf
Carnegie Mellon University - Tepper School of Business

YEQT 2014, Eindhoven
Outline

- Introduction & Motivation
- Model & methods
- Results
- Conclusions
Assemble-to-order (ATO) systems

- Large product variety managed by assembling the *products*, on demand, from multiple *components*
  - Broad assortment of products, yet cost efficient.
  - Good service: assembly is fast.
  - E.g. Dell (Kapuscinski et al., 2004), IBM (Swaminathan and Tayur 1998; Cheng et al. 2002), . . .
- Additional motivation: *any situation* where SKUs of different types are needed together to attain performance:
  - Multiple spare parts and/or tools in a maintenance task, e.g. Fokker (v J, Dollevoet, Dekker, 2011), Philips Healthcare (Kampstra, 2012), ASML (Vliegen; 2009) . . .
  - Multiple products in single customer demand, e.g. online retailers such as Amazon (Xu et al., 2009), . . .
ATO systems

ATO production: Components Products

Inventory of multiple SKUs Multiple SKUs demanded *together*
Controlling ATO systems

- Proper control is important to attain the cost efficiency promised by the ATO production system.
- Two related control problems:
  1. Controlling replenishment/production of each component.
  2. Controlling allocation of components to product demands.
- Companies need to control production and allocation of hundreds/thousands of components and products.
**Incomplete review**

1. Optimization of component **base-stock levels** ($(s, S)$ policies) under **FCFS allocation** (with committing):
   - Approximate optimization.
   - **SP formulation**: Swaminathan and Tayur (1998), Akçay and Xu (2004), Huang and de Kok (2011)
     - For positive leadtimes, SPs give rise to **non-convex optimization problem**.
     - Solved using “big-M” method. Thus limited to small problems.
Incomplete review

2 Optimality results on non-FCFS and coordinated replenishment, for special cases:

- No-holdback allocation for base-stock inventory control (special BOM, e.g. generalisations of W-model): Song and Zhao (2009), Lu et al. (2010)
- SP based analysis (W/N-model): Doğru et al. (2010), Lu et al. (2012), Reiman and Wang (2012);
Companies use pragmatic approaches:

- Independent base-stock ($(s, S)$) policies. Heuristics to determine parameters:
  - Ignore simultaneous stock-outs of multiple components (ISS):

  \[
  \text{Product } i \text{ wait} = \max_{j \in \mathcal{J}^i} (\text{Wait for component } j) \\
  \approx \sum_{j \in \mathcal{J}^i} (\text{Wait for component } j)
  \]

  e.g. IBM (Cheng et al.; 2002), Dell (Kapuscinski et al. ;2004), Fokker (vJ Dollevoet Dekker; 2011)

- Simple allocation rules: FCFS (commit inventory), FRFS (no-holdback)

- Because ISS and FCFS/FRFS are easy to implement.

- **But what is the performance of these approaches?**
Research questions

1. How to find provably (near-) optimal base-stock levels for large-scale ATO systems under FCFS allocation?
   - What can be achieved with FCFS and independent base-stock (no coordination)?

2. What are the costs of ignoring simultaneous stock-outs (ISS) while optimizing inventory?

3. What are the costs of using FCFS instead of optimal allocation in industrial-scale ATO systems?
   - How much further improvement if moving away from FCFS?
Ease of exposition

- **Continuous review**, multiple products ($\mathcal{I}$) and components ($\mathcal{J}$).
- Independent base-stock policies $s_j$ for components $j \in \mathcal{J}$.
- Vector of base-stock policies $\bar{s}$.
- Costs for on-hand component inventory and product demand back-orders:

  $$C = \sum_{j \in \mathcal{J}} h_j \mathbf{E}H_j(\bar{s}) + \sum_{i \in \mathcal{I}} b^i \mathbf{E}B^i(\bar{s})$$

- Poisson demand for products ($\lambda^i$)
- Deterministic replenishment leadtimes for components ($l_j$).
- Deterministic $0-1$ BOM:
  - $\mathcal{J}^i$ components used in product $i$.
  - $\mathcal{I}_j$ products in which component $j$ is used.
Lu and Song (2005) show:

$$C = \sum_{j \in J} h_j s_j + \sum_{i \in I} \tilde{b}_i \mathbb{E}B^i(\bar{s}) - D$$

where $\tilde{b}_i = b_i + \sum_{j \in J^i} h_j$, and $D$ is constant.

Use exact computation of $\mathbb{E}B^i(\bar{s})$ to optimize?
- No: computation only tractable for $|J^i| \leq 3$.
- No: non-linear, non-separable optimization; dimension of search space.

Instead, we develop an SP formulation, and sampling procedures.

$\mathbb{E}B^i(\bar{s}) = \lambda^i \mathbb{E}W^i(\bar{s})$, where $W^i$ is the random waiting time incurred by an arbitrary demand of product $i$.

Use tagged job, consider arbitrary arrival of demand for product $i$, and determine $W^i$. 
Tagged job approach

Demand for product of type $i \in \mathcal{I}$ ($t := 0$)

$W^i = \max_{j \in \mathcal{J}^i} W^i_j(s_j)$

$I = \text{"Demand" for } j \in \mathcal{J}^i$

$I = \text{"Demand" for } j' \in \mathcal{J}^i$
SP formulation under FCFS

- \( x_{js} \) indicates that base-stock level \( s \) is used for part \( j \).
- Two-stage stochastic program:

\[
\min \sum_{j \in J} \sum_{s=0}^{s_j^u} h_j s x_{js} - D + \sum_{i \in I} \tilde{b}^i \lambda^i w^i
\]

\[
w^i = \mathbf{E} \max_{j \in J^i} \left( \sum_{s=0}^{s_j^u} x_{js} W_j^i(s) \right)
\]

\[
\sum_{s \in C_j} x_{js} = 1 \quad j \in J
\]

\[
x_{js} \in \{0, 1\} \quad j \in J, s \in \{0, \ldots, s_j^u\}
\]

- This SP can be efficiently solved using an SAA approach.
- To find feasible solutions and asymptotic lower bounds.
SAA approach

- Sample consists of $N^i$ independent realizations of arrivals of each product $i$: $\xi^i_n = \{W^i_j(s)(\xi^i_n)|j \in J^i, s \in \{0, \ldots, s^u_j\}\}$.  
- SAA of SP for sample is given by:

$$\min \sum_{j \in J} \sum_{s=0}^{s^u_j} h^j s x_{js} - D + \sum_{i \in I} \frac{1}{|N^i|} \tilde{b}^i \lambda^i \sum_{n \in N^i} v^i_n,$$

$$v^i_n \geq \sum_{s=0}^{s^u_j} x_{js} W^i_j(s)(\xi^i_n), \quad i \in I, j \in J^i, n \in N^i,$$

$$\sum_{s=0}^{s^u_j} x_{js} = 1, \quad j \in J,$$

$$v^i_n \in \mathcal{R}, \quad i \in I, n \in N^i,$$

$$x_{js} \in \{0, 1\}, \quad j \in J, s \in \{0, \ldots, s^u_j\}.$$
Scalability

| Size $|\mathcal{J}| = |\mathcal{I}|$ of test case | Computation time (Hrs.) |
|-----------------------------------------------|-------------------------|
|                                               | Tagged-Job approach     | Akçay and Xu (2004) $^1$ | Huang and De Kok (2011) |
| 5                                             | 0.01                    | 0.11                      | 1.2                      |
| 10                                            | 0.03                    | 0.11                      | 4.5                      |
| 15                                            | 0.04                    | 1.5                       | 40.1                     |
| 20                                            | 0.07                    | 88.8                      | -                        |
| 50                                            | 0.17                    | -                         | -                        |
| 100                                           | 0.4                     | -                         | -                        |
| 200                                           | 0.88                    | -                         | -                        |
| 300                                           | 1.45                    | -                         | -                        |

Our tagged-job approach is the first scalable algorithm for determining base-stock levels in FCFS ATO systems.

$^1$Ignores committed inventories.
Lower bound under optimal allocation

For better insights into non-FCFS policies: lower bound on the costs of the best base-stock policy under optimal allocation.

- Idea 1: under *any* allocation rule, on hand inventory at time $t := 0$ must be bigger than 0. Minimize back-orders at 0, subject to only this constraint:
  - Doğru et al, 2010: two-stage SP for equal leadtime case, LB on optimal policy.
  - Reiman and Wang, 2012: n-stage SP for unequal leadtime case, LB on optimal policy.
  - Employed to find the optimal replenishment + allocation policy for W-model under cost symmetry.
Lower bound under optimal allocation

- We: under base-stock policy, on-hand inventory at 0 equals
  \[ s_j + (\text{back-orders at } -l_j) - (\text{satisfied demands during } (-l_j, 0]) \]

- We use this relation to develop an SP lower bound on the costs of the optimal base-stock policy under optimal allocation.

- Two-stage SP, also for the non-equal leadtime case.

- We propose SAA algorithm to solve for general ATO systems.
Case: Assembly of products of multiple families

- Many OEMs divide their products into product families (e.g. medical equipment or wafer steppers, see De Kok (2003)).
- Developed cases with this property:
  - 3 product families, 20 common components, and $3 \times 8$ (more expensive) family specific components.
  - Products in each family are assembled from 5 common components and $n_s \in \{1, \ldots, 8\}$ family specific components.
- Choose demand rates, Holding costs and leadtimes randomly. Leadtimes uniformly on $[1 - \delta_l, 1 + \delta_l]$.
- $n_s$ and $\delta_l$ are varied.
Two additional parameters - news-vendor fractile $f$ and asymmetry $\delta_b$ - for setting back-order penalties:

$$b^i = \frac{f}{1 - f} h^i (1 + \delta_b x^i)$$

Here, $h^i = \sum_{j \in \mathcal{J}_i} h_j$.

And $x^i \in \{-0.5, 0.0, 0.5\}$ governs whether the product is a low, medium or high criticality.

While setting $x^i$, we use two cases, representing within product family and between product family asymmetry.
Three policies for setting base-stock levels: ignoring simultaneous stockouts “iss”, our tagged-job approach: “spfc”; the base-stock levels from the lower bound on optimal allocation “drw”.

Three allocation rules:
- FCFS: commits inventory, denoted “-fc”
- FRFS: No holdback/committing, clears back-orders FCFS “-fr”
- No holdback, clear back-orders with highest modified penalty costs $\tilde{b}_i$ first: “-pr”.

Two lower bound estimates: a lower bound under FCFS allocation “lb-fc”, and a lower bound for base-stock policies under optimal allocation “lb-all”.

We report relative differences with the lower bound under FCFS, for example:

\[
\text{iss-fc}\% := \frac{\text{iss-fc} - \text{lb-fc}}{\text{lb-fc}} \times 100\%.
\]
Multiple family assembly : ISS-FC policies

Bars represent 99.7% confidence intervals
Multiple family assembly: Tagged-job approach

\[ f: \text{newsvendor fractile} \]

Relative performance (%) for different values of \( n_s \) and \( \delta l \):
- \( n_s = 7, \delta l = 0 \)
- \( n_s = 7, \delta l = 1 \)
- \( n_s = 6, \delta l = 0 \)
- \( n_s = 6, \delta l = 1 \)
- \( n_s = 4, \delta l = 0 \)
- \( n_s = 4, \delta l = 1 \)

Diagram shows the performance for different fractiles and parameters.
Multiple family assembly: non-FCFS

Between product family assymmetry ($n_s = 6, \delta_l = 1, \delta_b = 1$):

```
<table>
<thead>
<tr>
<th>f</th>
<th>spfc-fr%</th>
<th>spfc-pr%</th>
<th>spfc-fc%</th>
<th>drw-fr%</th>
<th>drw-pr%</th>
<th>lb-fc%</th>
<th>lb-all%</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.85</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.95</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.97</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.99</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

$f$: newsvendor fractile
Within product family asymmetry ($n_s = 6, \delta_l = 1, \delta_b = 1$):
Repair shop case

- Based on data from aircraft component repair shop (Fokker Aerospace).
- 3 repair types ($\mathcal{I} = \{a, b, c\}$):
  - arrival rates 0.13, 0.10 and 0.35.
- 110 spare parts:
  - holding cost ($h$) and leadtime ($l$) given.
  - Each part used in each repair type with given probability ($p_a, p_b, p_c$).
- Define holding costs:
- Define the backorder costs for repairs of type $i \in \{a, b, c\}$ as
  \[ b^i = h^i \frac{f}{(1 - f)}(1 + x^i) \]
  with
  - $x^a = 0.5\delta b$, $x^b = 0$, $x^c = -0.5\delta b$
  - $h^i := \sum_{j=1}^{110} p^i_j h_j$. 
### Repair shop data (excerpt):

<table>
<thead>
<tr>
<th>$h$</th>
<th>$l$</th>
<th>$p_a$</th>
<th>$p_b$</th>
<th>$p_c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>341</td>
<td>55</td>
<td>2.3%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>270</td>
<td>55</td>
<td>2.3%</td>
<td>2.9%</td>
<td>-</td>
</tr>
<tr>
<td>270</td>
<td>55</td>
<td>-</td>
<td>-</td>
<td>8.3%</td>
</tr>
<tr>
<td>249</td>
<td>41</td>
<td>4.7%</td>
<td>2.9%</td>
<td>4.6%</td>
</tr>
<tr>
<td>240</td>
<td>41</td>
<td>2.3%</td>
<td>5.7%</td>
<td>2.8%</td>
</tr>
<tr>
<td>213</td>
<td>55</td>
<td>16.3%</td>
<td>5.7%</td>
<td>-</td>
</tr>
<tr>
<td>175</td>
<td>34</td>
<td>2.3%</td>
<td>2.9%</td>
<td>0.9%</td>
</tr>
<tr>
<td>162</td>
<td>55</td>
<td>-</td>
<td>5.7%</td>
<td>-</td>
</tr>
<tr>
<td>156</td>
<td>45</td>
<td>7%</td>
<td>-</td>
<td>4.6%</td>
</tr>
<tr>
<td>156</td>
<td>31</td>
<td>-</td>
<td>-</td>
<td>0.9%</td>
</tr>
<tr>
<td>128</td>
<td>28</td>
<td>-</td>
<td>-</td>
<td>0.9%</td>
</tr>
<tr>
<td>123</td>
<td>16</td>
<td>2.3%</td>
<td>2.9%</td>
<td>1.8%</td>
</tr>
<tr>
<td>123</td>
<td>13</td>
<td>4.7%</td>
<td>-</td>
<td>2.8%</td>
</tr>
<tr>
<td>105</td>
<td>37</td>
<td>-</td>
<td>-</td>
<td>1.8%</td>
</tr>
<tr>
<td>105</td>
<td>37</td>
<td>-</td>
<td>-</td>
<td>14.7%</td>
</tr>
<tr>
<td>105</td>
<td>17</td>
<td>-</td>
<td>-</td>
<td>0.9%</td>
</tr>
<tr>
<td>105</td>
<td>17</td>
<td>-</td>
<td>-</td>
<td>0.9%</td>
</tr>
</tbody>
</table>

### Additional Data

<table>
<thead>
<tr>
<th>$h$</th>
<th>$l$</th>
<th>$p_a$</th>
<th>$p_b$</th>
<th>$p_c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>56</td>
<td>21</td>
<td>4.7%</td>
<td>5.7%</td>
<td>1.8%</td>
</tr>
<tr>
<td>52</td>
<td>7</td>
<td>7%</td>
<td>8.6%</td>
<td>6.4%</td>
</tr>
<tr>
<td>50</td>
<td>19</td>
<td>-</td>
<td>-</td>
<td>9.2%</td>
</tr>
<tr>
<td>50</td>
<td>21</td>
<td>2.3%</td>
<td>2.9%</td>
<td>0.9%</td>
</tr>
<tr>
<td>50</td>
<td>6</td>
<td>-</td>
<td>-</td>
<td>33%</td>
</tr>
<tr>
<td>50</td>
<td>6</td>
<td>-</td>
<td>7%</td>
<td>8.3%</td>
</tr>
<tr>
<td>50</td>
<td>19</td>
<td>18.6%</td>
<td>31.4%</td>
<td>20.2%</td>
</tr>
<tr>
<td>50</td>
<td>17</td>
<td>2.3%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>46</td>
<td>21</td>
<td>-</td>
<td>2.9%</td>
<td>-</td>
</tr>
<tr>
<td>46</td>
<td>17</td>
<td>4.7%</td>
<td>-</td>
<td>4.6%</td>
</tr>
<tr>
<td>44</td>
<td>12</td>
<td>-</td>
<td>-</td>
<td>0.9%</td>
</tr>
<tr>
<td>43</td>
<td>12</td>
<td>4.7%</td>
<td>-</td>
<td>0.9%</td>
</tr>
<tr>
<td>41</td>
<td>50</td>
<td>2.3%</td>
<td>8.6%</td>
<td>-</td>
</tr>
<tr>
<td>41</td>
<td>23</td>
<td>2.3%</td>
<td>8.6%</td>
<td>-</td>
</tr>
<tr>
<td>41</td>
<td>3</td>
<td>4.7%</td>
<td>2.9%</td>
<td>5.5%</td>
</tr>
<tr>
<td>39</td>
<td>13</td>
<td>-</td>
<td>-</td>
<td>0.9%</td>
</tr>
<tr>
<td>39</td>
<td>12</td>
<td>-</td>
<td>2.9%</td>
<td>0.9%</td>
</tr>
</tbody>
</table>
$\delta b = 1, \lambda = 2.0$: 

![Graph showing relative performance (%) vs. newsvendor fractile.](image-url)
<table>
<thead>
<tr>
<th>Policy</th>
<th>Benchmark</th>
<th>Params</th>
<th>Cost increase per NV fractile</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.99</td>
</tr>
<tr>
<td>ISS</td>
<td>$lb-fc$</td>
<td>Dem</td>
<td>L: 1-2%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>corr.</td>
<td>M: 5-8%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>H: 12-30%</td>
</tr>
<tr>
<td>SPFC</td>
<td>$lb-fc$</td>
<td>-</td>
<td>Always &lt; 0.5%</td>
</tr>
<tr>
<td>FCFS (lb-fc)</td>
<td>$lb-opt$</td>
<td>P. asym.</td>
<td>L: 4-8%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>H: 8-18%</td>
</tr>
<tr>
<td>FCFS (lb-fc)</td>
<td>NHB-PR</td>
<td>P. Asym.</td>
<td>L: 3-4%</td>
</tr>
<tr>
<td></td>
<td>(spfc/drw)</td>
<td></td>
<td>H: 5-8%</td>
</tr>
</tbody>
</table>

Opportunity
Conclusion

- We developed the first algorithm that can find close to optimal solutions for large ATO systems, without using the ISS assumption.
- We generated insights for the performance of control policies for realistic cases.
- We found that ISS performs well as long as demand correlation during leadtime is low, or if NV fractiles are high.
- But can also perform poorly if demand correlation during leadtime is high and NV fractiles are low. In those cases, consider SPFC.
- FCFS performs surprisingly well, even if there is significant penalty asymmetry.
- Simple no-holdback policies slightly outperform the best FCFS policy.