The Joint Replenishment Problem
Optimal Policy And Exact Evaluation Method

Stefan Creemers
Robert Boute
(2021 INFORMS Annual Meeting)
Introduction

• The Joint Replenishment Problem (JRP) is a well-known problem in the ORMS literature
• 2,060 hits on Google Scholar
• General idea:
  – You keep several Stock Keeping Units (SKUs) in inventory.
  – For each SKU $i$, you incur a holding cost $h_i$ and face a Poisson demand with rate parameter $\lambda_i$.
  – You can replenish the inventory of an SKU by issuing an order that has major order cost $K$. For each SKU $i$ included in the order, you incur minor order cost $k_i$.
• Million-dollar question: how do we coordinate orders such that holding and order costs are minimized?
Introduction


• Finding the optimal control policy is intractable, even for problems with only two SKUs

• Examples of tractable policies:
  – Periodic policy; see e.g., Atkins & Iyogun (1987) and Viswanathan (1997 & 2007)
  – Can-order policy
Introduction


• Finding the optimal control policy is intractable, even for problems with only two SKUs

• Examples of tractable policies:
  – Periodic policy; see e.g., Atkins & Iyogun (1987) and Viswanathan (1997 & 2007)
  – Can-order policy
Can-order policy

- Introduced by Balintfly (1964)
- For each SKU $i$ there are three parameters:
  - Order-up-to level $S_i$
  - Can-order level $c_i$
  - Reorder point $s_i$
- If the inventory of one of the SKUs hits the reorder point, a replenishment order is triggered, and any other SKU that has inventory below/at the can-order level joins the order
- The exact cost of a can-order policy can be determined using a Continuous-Time Markov Chain (CTMC)
- However, for systems with more than a few SKUs, the CTMC becomes too big, and we can no longer determine the best can-order policy (curse of dimensionality!)
Decomposition approach

- Introduced by Silver E.A. (1974), and further refined by Federgruen et al. (1984)
- Can be used to obtain a “good” can-order policy, even for large problems with many SKUs
- The decomposition approach decomposes the JRP into single-item problems that are solved iteratively:
  - For each SKU $i$, find the best can-order parameters $(S_i, c_i, s_i)$ in a single-item system where the replenishment orders of other SKUs are captured using so-called “special replenishment opportunities” that arrive with rate $\mu_i$.
  - Given the updated can-order policy for SKU $i$, determine the new rate of special replenishment opportunities $\mu_j$ for all other SKUs $j \neq i$.
  - Repeat this procedure for each SKU until the can-order policy itself convergences.
- Although the decomposition approach resolves the curse of dimensionality, there are some drawbacks:
  - It is a heuristic procedure (the single-item problem for SKU $i$ ignores the interaction of SKUs $j \neq i$; all interaction is captured by a single parameter $\mu_i$).
  - It approximates the cost of a single-item system using a closed-form expression. As a result, we need to simulate the can-order policy in order to obtain its real cost. In addition, to determine whether one can-order policy is better than another, we base ourselves on approximate costs (that may differ substantially from the real cost).
Main contributions

• New, exact method to determine the cost of a JRP that partially solves the curse of dimensionality
New, exact method

• In a traditional CTMC approach, a state is defined as a tuple \((I_1, ..., I_N)\) (with \(I_i\) the inventory of SKU \(i\), and \(N\) the number of SKUs). For a given can-order policy, the number of states is given by \(\prod_{i=1}^{N}(S_i - s_i)\). Even for problems with only a few SKUs, the CTMC can no longer be analyzed.

• We propose a new approach that uses a Discrete-Time Markov Chain (DTMC) that models transitions between so-called “initial states”; states in which we end up after an order has been triggered. By considering only initial states, we can reduce the number of states in our DTMC to \(\sum_{i=1}^{N} \prod_{j \neq i}(S_i - c_i)\).

• The reduction in the number of states can be significant:

<table>
<thead>
<tr>
<th>Number of states required for analyzing the best can-order policy for the Federgruen instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example problem</td>
</tr>
<tr>
<td>Traditional CTMC</td>
</tr>
<tr>
<td>New DTMC</td>
</tr>
</tbody>
</table>

• This huge reduction in the number of states allows us to analyze JRP policies of larger problems with several SKUs.

• In addition, we can easily extend our method (compound Poisson demand, lead time, backlog, lost sales...) without increasing the number of states.
Main contributions

- New, exact method to determine the cost of a JRP that partially solves the curse of dimensionality
- Characterization of the optimal JRP policy
Optimal JRP policy

• The optimal policy has a can-order structure; if a set of SKUs joins the order triggered by SKU $i$, they will do so if their inventory level is at/below a given can-order level. The can-order level (and the order-up-to level) of the SKUs that join the order depends on the inventory levels of the SKUs that do not join the order.

• Two important implications:
  – The can-order policy is a logical heuristic; it adopts the structure of the optimal policy.
  – However, the can-order policy assumes a single can-order level for each SKU independent of the inventory levels of the SKUs that do not join the order ➔ if the number of SKUs increases, the optimality gap is expected to increase as well!
Main contributions

• New, exact method to determine the cost of a JRP that partially solves the curse of dimensionality
• Characterization of the optimal JRP policy
• Introduction of a new, generalized can-order policy
Generalized can-order policy

- Using the insights of the optimal JRP policy, the can-order policy can be generalized using a greedy procedure:
  - Start from the best can-order policy.
  - For each combination of inventory levels of SKUs that do not join the order, evaluate whether it is beneficial to alter the can-order level (and/or order-up-to level) of SKUs that do join the order.
  - Repeat until no further improvement can be found.

- After applying this to the Federgruen instances, we get:

<table>
<thead>
<tr>
<th>Expected cost of can-order policy and generalized can-order policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example problem</td>
</tr>
<tr>
<td>Best can-order policy</td>
</tr>
<tr>
<td>Generalized can-order policy</td>
</tr>
</tbody>
</table>

- The difference is not substantial, however, we expect the gap to increase if the number of SKUs increases!
Main contributions

• New, exact method to determine the cost of a JRP that partially solves the curse of dimensionality
• Characterization of the optimal JRP policy
• Introduction of a new, generalized can-order policy
• Generalization of the decomposition approach
Generalized decomposition approach

- Our exact method can analyze problems with several SKUs. Therefore, we can generalize the decomposition approach, and now also decompose the JRP into double-item and triple-item problems (next to single-item problems).
- In addition, rather than using an approximate (closed-form) cost function, we use our method to analyze the exact cost of the single/double/triple-item problems.

<table>
<thead>
<tr>
<th>Example problem</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decomposition approach (approximation)</td>
<td>88.71</td>
<td>89.98</td>
<td>71.53</td>
</tr>
<tr>
<td>Decomposition approach (exact)</td>
<td>81.03</td>
<td>83.62</td>
<td>68.52</td>
</tr>
<tr>
<td>Generalized decomposition (single item)</td>
<td>80.07</td>
<td>82.66</td>
<td>68.70</td>
</tr>
<tr>
<td>Generalized decomposition (double item)</td>
<td>78.10</td>
<td>82.16</td>
<td>68.04</td>
</tr>
<tr>
<td>Generalized decomposition (triple item)</td>
<td>77.97</td>
<td>81.27</td>
<td>67.96</td>
</tr>
<tr>
<td>Best can-order policy</td>
<td>77.51</td>
<td>80.87</td>
<td>67.80</td>
</tr>
</tbody>
</table>
Main contributions

• New, exact method to determine the cost of a JRP that partially solves the curse of dimensionality
• Characterization of the optimal JRP policy
• Introduction of a new, generalized can-order policy
• Generalization of the decomposition approach