Models and algorithms for the dynamic-demand joint replenishment problem

Boctor, F. F., Laporte, G., and Renaud, J.

IJPR 2004 Vol. 42 No. 13

Doh, Hyoung Ho 2016. 8. 18
1. Introduction
2. Problem Description
3. Heuristics
4. Computational experiment
5. Conclusions
1. Introduction

Dynamic Demand Joint Replenishment Problem (DJRP)

- $S_i$: common ordering cost
- $s_{it}$: individual ordering cost

\[x_{it}, x_{i+1,t}, \ldots, x_{nt}\]

\[d_{it}, d_{i+1,t}, \ldots, d_{nt}\]

1. Multiple products
2. No capacity restriction
3. When and how mulch to order for each item $i$ and for each period $t$
4. 2 kinds of setup occur, i.e., common ordering cost, individual ordering cost
2. Problem Description

- **Objective function**
  Minimize the total cost (Common ordering, individual ordering, and inventory holding costs)

- **Decision variables**
  1. When and how mulch to order for each item $i$ and for each period $t$
  2. Ordering costs (2 types)
  3. Inventory level for each item type at the end of period

- **Assumptions**
  1. No initial inventory in any level
  2. Terminal inventory is zero
  3. Backlogging is not allowed
  4. All parameters are to be nonnegative
  5. No lead time

- **Constraints**
  1. Inventory balance
  2. Setups

- **Approach**
  Formulation, Heuristics
2. Problem Description

Notation

Indices

\( i \) \quad \text{items, } i = 1, 2, \ldots, n
\( t \) \quad \text{periods, } t = 1, 2, \ldots, T

Parameters

\( S_i \) \quad \text{common ordering cost at period } t
\( s_{it} \) \quad \text{individual ordering cost for item type } i \text{ at period } t
\( h_{it} \) \quad \text{unit inventory holding cost for item type } i \text{ during period } t
\( d_{it} \) \quad \text{demand for item type } i \text{ for period } t
\( M \) \quad \text{sufficiently large number}

Decision Variables

\( x_{it} \) \quad \text{replenishment quantity of item type } i \text{ at the beginning of period } t
\( y_{it} \) \quad \text{binary variables } = 1 \text{ if and only if item type } i \text{ is replenished at the beginning of period } t
\( I_{it} \) \quad \text{inventory level of item type } i \text{ at the end of period } t
\( z_{it} \) \quad \text{binary variables } = 1 \text{ if an order is placed for period } t
\( B_{it} \) \quad \text{sum of demands for item } i \text{ from period } t \text{ to last period } T
2. Problem Description

Formulation – Echelon stock (ES) : Federgruen and Tzur (1999)

Objective Function

\[(DJRP1) \min \sum_{t=1}^{T} \left[ S_t z_t + \sum_{i=1}^{N} \{s_{it} y_{it} + h_{it} I_{it}\} \right] \tag{1}\]

s.t

\[I_{i,t-1} + x_{it} - I_{it} = d_{it} \quad (i = 1, \ldots, n, \ t = 1, \ldots, T) \tag{2}\]

\[x_{it} \leq M y_{it} \quad (i = 1, \ldots, n, \ t = 1, \ldots, T) \tag{3}\]

\[\sum_{i=1}^{n} y_{it} \leq n z_t \quad (t = 1, \ldots, T) \tag{4}\]

\[I_{it} \geq 0 \quad (i = 1, \ldots, n, \ t = 1, \ldots, T) \tag{5}\]

\[x_{it} \geq 0 \quad (i = 1, \ldots, n, \ t = 1, \ldots, T) \tag{6}\]

\[y_{it} = 0 \text{ or } 1 \quad (i = 1, \ldots, n, \ t = 1, \ldots, T) \tag{7}\]

\[z_t = 0 \text{ or } 1 \quad (t = 1, \ldots, T) \tag{8}\]
2. Problem Description

Properties

- Property 1: Any optimal DJRP solution is such that: $x_{i_1}^* \times I_{i_{t-1}}^* = 0$ ($i = 1, \ldots, n; t = 1, \ldots, T$). In other words, if item type $i$ is replenished at the beginning of period $t$, it does not pay to hold this item type in stock during period $t-1$ (Wagner and Within 1958).

- Property 2: The optimal order $x_{i,t}^*$ takes one of the values: $d_{i,t}, d_{i,t} + d_{i,t+1}, \ldots, \sum_{q=1}^{T} d_{i,q}$.

- Property 3: The optimal inventory level $I_{i_{t-1}}^*$ takes one of the values: $0, d_{i,t}, d_{i,t} + d_{i,t+1}, \ldots, \sum_{q=1}^{T} d_{i,q}$.

- Property 4: If $d_{i,q} \sum_{r=t}^{q-1} h_{i,r} > S_q + s_{i,q}$ for any $q > t$, then it is not optimal to replenish the demand of item type $i$ for period $q$ at the beginning of period $t$. 

### 2. Problem Description

**Formulation – second model**

*Objective Function*

\[
(DJR P 2) \min \sum_{t=1}^{T} \left[ S_i z_t + \sum_{i=1}^{n} \sum_{q=t}^{T} c_{itq} w_{itq} \right]
\]  

\[
\sum_{q=1}^{t} \sum_{r=t}^{T} w_{iqr} = 1 \quad (i = 1, \ldots, n; \ t = 1, \ldots, T)
\]  

\[
\sum_{i=1}^{n} \sum_{q=t}^{T} w_{iiq} \leq nz_t \quad (t = 1, \ldots, T)
\]  

\[
w_{itq} = 0 \text{ or } 1 \quad (i = 1, \ldots, n; \ t = 1, \ldots, T; \ q = 1, \ldots, T)
\]  

\[
z_t = 0 \text{ or } 1 \quad (t = 1, \ldots, T).
\]

\[W_{itq} = 1 \text{ if and only if the replenishment order of item type } i \text{ at the beginning of period } t \]

\[\text{covers the demand for this item type for all periods until period } q\]

\[w_{itq} = 1 \iff x_{it} = \sum_{r=t}^{q} d_{ir}.\]
2. Problem Description

Formulation – second model

Objective Function

\[(DJRP3) \min \sum_{i=1}^{T} S_{it}z_{it} + \sum_{i=1}^{n} \sum_{r=1}^{T} s_{it}u_{itr} + \sum_{i=1}^{n} \sum_{t=2}^{T} \sum_{q=1}^{t-1} \left( \sum_{r=q}^{t-1} h_{ir} \right) d_{it}u_{iq}\]

\[\sum_{q=1}^{t} u_{iq} = 1 \quad (i = 1, \ldots, n; \ t = 1, \ldots, T) \quad (14)\]

\[\sum_{i=1}^{n} u_{itt} \leq nz_t \quad (t = 1, \ldots, T) \quad (15)\]

\[u_{itq} \leq u_{itt} \quad (i = 1, \ldots, n; \ t = 1, \ldots, T - 1; \ q = t + 1, \ldots, T) \quad (16)\]

\[u_{itq} = 0 \text{ or } 1 \quad (i = 1, \ldots, n; \ t = 1, \ldots, T; \ q = 1, \ldots, T) \quad (17)\]

\[z_t = 0 \text{ or } 1 \quad (t = 1, \ldots, T). \quad (18)\]

\[u_{itq} = 1 \text{ if and only if the demand of item } i \text{ for period } q \text{ is included in the replenishment made at the beginning of period } t. \quad (19)\]
3. Summary of some classical heuristics

Fogarty and Barringer heuristic (1987)

\[ f_t = \min_{q \leq t} \{ f_{q-1} + c_{qt} \} \quad (t = 2, \ldots, T) \]

\[ f_1 = S_1 z_1 + \sum_{i=1}^{n} s_{i1} y_{i1} \]

\[ c_{qt} = S_q z_q + \sum_{i=1}^{n} \left\{ s_{iq} y_{iq} + \sum_{r=q+1}^{t} d_{ir} \left( \sum_{k=q}^{r-1} h_{ik} \right) \right\} \]

\[ z_q = 1 \text{ if } \sum_{i=1}^{n} \sum_{r=q}^{t} d_{ir} > 0, \text{ and } z_q = 0 \text{ otherwise} \]

\[ y_{iq} = 1 \text{ if } \sum_{r=q}^{t} d_{ir} > 0, \text{ and } y_{iq} = 0 \text{ otherwise}. \]
3. Summary of some classical heuristics

Greedy add heuristic (1994)

Step 1. Initialization: Set \( x_{i1} = \sum_{t=1}^{T} d_{it} \) \((i = 1, \ldots, n)\), and \( P = \{2, \ldots, T\} \).

Step 2. Determination of best saving: for each \( t \in P \), let \( q_{t}^- \) be the latest replenishment period before \( t \) and \( q_{t}^+ \) be the period of the first replenishment after \( t \). If there is no replenishment after \( t \), set \( q_{t}^+ = T + 1 \). Compute the potential saving value:

\[
g_t = \left\{ \sum_{i=1}^{n} \left( \sum_{r=q_t^-}^{t} h_{ir} \right) \left( \sum_{r=t}^{q_t^+ - 1} d_{ir} \right) \right\} - \left\{ S_t + \sum_{i=1}^{n} s_{it} y_{it} \right\}.
\]

\[y_{it} = 1 \text{ if } \sum_{r=t}^{q_t^+ - 1} d_{ir} > 0, \text{ and } y_{it} = 0 \text{ otherwise.}\]

Step 3. Add replenishment or stop: Remove from \( P \) all values of \( t \) for which \( g_t < 0 \). If \( P = \emptyset \), stop. Otherwise, make a replenishment at period \( t^* \) yielding the maximum saving \( g_t \) and accordingly reduce the replenishment made at period \( q_t^- \). Go to Step 2.
3. Summary of some classical heuristics

Greedy drop heuristic

Step 1. Initialization: Set \( x_{it} = d_{it} \) for \( i = 1, \ldots, n \) and \( t = 1, \ldots, T \). Set \( R = \{1, \ldots, T\} \).

Step 2. Determination of best saving: For each \( t \in R \), compute the potential savings:

\[
g_t = \left\{ S_t + \sum_{i=1}^{n} s_{it} y_{it} \right\} - \left\{ \sum_{i=1}^{n} \left( \sum_{r=q_i}^{t} h_{ir} \right) \left( \sum_{r=t}^{q_i-1} d_{ir} \right) \right\},
\]

\[
y_{it} = 1 \text{ if } \sum_{r=t}^{q_i-1} d_{ir} > 0, \text{ and } y_{it} = 0 \text{ otherwise}.
\]

Step 3. Drop replenishment or stop: Remove from \( R \) all values of \( t \) for such that \( g_t < 0 \). If \( R = \emptyset \), stop. Otherwise, cancel the replenishment at period \( t^* \) yielding the maximum saving \( g_t \) and accordingly increase the replenishment made at \( q_{it} \). Go to Step 2. Again, reducing \( R \) in Step 3 is justified by the fact that \( g_t \) cannot increase during the subsequent iterations of the algorithm.
3. Summary of some classical heuristics

Extended silver-meal heuristic  \( \text{SM} \) (1973)

\[
SM_t(s_q) = \left[ s_q + \sum_{r=q+1}^{t} \left( d_r \sum_{u=q}^{r-1} h_u \right) \right] / (t - q + 1), \quad \text{if } t > q,
\]

\[
SM_t(s_q) = s_q, \quad \text{if } t = q.
\]

**Step 1. Initialization:**

Set \( t = 1, \)

\( q_i = 1 \) \((i = 1, \ldots, n)\) (period of the last replenishment for item type \( i \))

\( q = 1 \) (period of the last replenishment)

\( B_q \) (set of item types included in the replenishment made at period \( q \))

\[
\bar{B}_q = \{1, \ldots, n\} \backslash B_q
\]

\[
SM_{it}(s_{iq_i}) = s_{it} (i = 1, \ldots, n).
\]
3. Summary of some classical heuristics

Extended silver-meal heuristic \[\text{SM (1973)}\]

**Step 2. Incrementation:** Set \( t = t + 1 \). Compute \( SM_{it}(s_{iq_i}) = [s_{iq_i} + \sum_{r=q_i+1}^{t} (d_{ir} \times \sum_{u=q_i}^{r-1} h_{iu})]/(t - q_i + 1) \) for \( i = 1, \ldots, n \), and define \( A_t \). For all \( i \in A_t \cap \overline{B}_q \), check whether \((\sum_{r=1}^{t} d_{ir})(\sum_{r=q_i}^{q-1} h_{ir}) > s_{iq}\). If this inequality holds, then replenish item type \( i \) at period \( q \). Add \( i \) to \( B_q \).

**Step 3. Test for replenishment at period \( t \):** For all \( i \in A_t \), compute \( \Delta_i \) such that \( SM_{i,t-1}(s_{iq_i} + \Delta_i) = SM_{it}(s_{iq_i} + \Delta_i) \). If \( \sum_{i \in A_t} \Delta_i \geq S_t \), go to Step 4. Otherwise, go to Step 2.

**Step 4. Replenishment at period \( t \):** Set \( q = t \), \( B_q = A_t \) and \( \overline{B}_q = \{1, \ldots, n\} \setminus B_q \). Also set \( q_i = t \) and \( SM_{it}(s_{iq_i}) = s_{iq_i} \) for all \( i \in A_t \). If \( t < T \), go to Step 2. Otherwise, stop.

\[
A_t = \left\{ i : SM_{i,t-1}(s_{i,q_i}) < SM_{it}(s_{iq_i}) \right\}
\]
\[
\{ i : SM_{i,t-1}(s_{i,q_i}) < SM_{it}(s_{iq_i}) \text{ and } \sum_{r=q_i+1}^{t-1} (d_{ir} \sum_{k=q_i}^{t-2} h_{ik}) > s_{iq_i} \}
\]
3. Summary of some classical heuristics

**Generalized part-period balancing heuristic  <==============  PP (1968)**

*Step 1. Initialization:* Set: \( t = 1, \ q = 1 \) and \( q_i = 1 \) for \( i = 1, \ldots, n \).

*Step 2. Determination of the candidate replenishment periods:* For all \( t > q \) and \( t \leq T \), determine the set \( C_t = \{ i : H_{it} = \sum_{r=q+1}^{t} d_r (\sum_{q=r}^{r-1} h_{iq}) > s_{it} \} \). Let \( u \) be the earliest period \( t \) for which \( \sum_{i \in C_t} (H_{it} - s_{it}) > S_t \). If no such period exists, stop.

*Step 3. Determination of the next replenishment period:* If \( \sum_{i \in C_u} (H_{iu} - s_{iu}) - S_u < S_{u-1} - \sum_{i \in C_{u-1}} (H_{i,u-1} - s_{i,u-1}) \), replenish all item type of \( C_u \) at period \( u \) and set \( q = u \) and \( q_i = u \) for all \( i \in C_u \). Otherwise, replenish all item types of \( C_{u-1} \) at period \( u - 1 \) and set \( q = u - 1 \) and \( q_i = u - 1 \) for all \( i \in C_{u-1} \). Go to Step 2.
3. Summary of some classical heuristics

**Generalized part-period balancing heuristic**<br>\( \text{PP} (1968) \)

**Step 1. Initialization:** Set: \( t = 1, q = 1 \) and \( q_i = 1 \) for \( i = 1, \ldots, n \).

**Step 2. Determination of the candidate replenishment periods:** For all \( t > q \) and \( t \leq T \), determine the set \( C_t = \{ i : H_{it} = \sum_{r=q_i+1}^{t} d_{ir} (\sum_{q=q_i}^{r-1} h_{iq}) > s_{it} \} \). Let \( u \) be the earliest period \( t \) for which \( \sum_{i \in C_t} (H_{it} - s_{it}) > S_t \). If no such period exists, stop.

**Step 3. Determination of the next replenishment period:** If \( \sum_{i \in C_u} (H_{iu} - s_{iu}) - S_u < S_{u-1} - \sum_{i \in C_{u-1}} (H_{i,u-1} - s_{i,u-1}) \), replenish all item type of \( C_u \) at period \( u \) and set \( q = u \) and \( q_i = u \) for all \( i \in C_u \). Otherwise, replenish all item types of \( C_{u-1} \) at period \( u - 1 \) and set \( q = u - 1 \) and \( q_i = u - 1 \) for all \( i \in C_{u-1} \). Go to Step 2.
3. Summary of some classical heuristics

New improvement heuristic

Step 1. Initial solution: Define the current solution as a DJRP feasible solution obtained by any heuristic.

Step 2. Solution perturbation: repeat the following operations $\lambda$ times. Choose $t$ randomly in $[1, T]$. If the current solution contains a replenishment at period $t$, cancel it and combine it with the previous replenishment. Otherwise, add a replenishment at period $t$ including all items contained in the previous replenishment which is then reduced accordingly.

Step 3. Solution improvement: Attempt to improve the perturbed solution by applying the greedy drop heuristic followed by the Silver–Kelle improvement heuristic. If the best known solution has improved, update it. Go to Step 2 until the following stopping criterion has been reached.

Stopping criterion

Steps 2 and 3 are applied as long as the best known solution has not improved for a set number $\theta$ of consecutive iterations. The present implementation used $\lambda = 3$ and $\theta = 6T$. Other values have been tried and these values seem to give better results.
4. Computational experiments

Environment
- MIP solver : CPLEX 8.0 with time limit 7200sec
- Pentium 3 core 2 Duo 1.26GHz

DATA : 720 instances
- \((N,T) = (10, 13), (10, 26), (20, 13), (20, 26)\)
- Subgroup : 6
- \(S = 1000\) for all \(t\)
- \(h_i = U[0.1, 0.6]\)

Performance measure
- Computational time for mathematical model
- \%GAP

\[
100\left(\frac{Z^*(.) - v(.)}{Z^*(.)}\right)
\]

\(Z^*(.)\) : optimal
\(v(.)\) : heuristic value
4. Computational experiments

Results

<table>
<thead>
<tr>
<th>Subgroup</th>
<th>$\alpha$</th>
<th>$\beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>0.5</td>
<td>6.0</td>
</tr>
<tr>
<td>S2</td>
<td>0.5</td>
<td>10.0</td>
</tr>
<tr>
<td>S3</td>
<td>1.0</td>
<td>6.0</td>
</tr>
<tr>
<td>S4</td>
<td>1.0</td>
<td>10.0</td>
</tr>
<tr>
<td>S5</td>
<td>2.0</td>
<td>6.0</td>
</tr>
<tr>
<td>S6</td>
<td>2.0</td>
<td>10.0</td>
</tr>
</tbody>
</table>

Table 1. Values of $\alpha$ and $\beta$.

$$\sum_{i=1}^{n} \frac{s_i}{S} = \alpha$$

$$d_{it} = 5\left[2\bar{X}\mu_{it}/5\right]$$

$$\mu_{it} = (s_i + 2\bar{X}S/n)/\beta h_i \quad [0, 1]$$
### 5. Computational experiments

Results: computational time for 3 formulations

<table>
<thead>
<tr>
<th>Subgroup</th>
<th>DJRP1</th>
<th>DJRP2</th>
<th>DJRP3</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>9.86</td>
<td>3.93</td>
<td>5.57</td>
</tr>
<tr>
<td>S2</td>
<td>18.20</td>
<td>3.85</td>
<td>6.63</td>
</tr>
<tr>
<td>S3</td>
<td>9.85</td>
<td>4.13</td>
<td>4.73</td>
</tr>
<tr>
<td>S4</td>
<td>15.88</td>
<td>3.99</td>
<td>8.05</td>
</tr>
<tr>
<td>S5</td>
<td>10.15</td>
<td>4.11</td>
<td>8.16</td>
</tr>
<tr>
<td>S6</td>
<td>18.24</td>
<td>3.97</td>
<td>5.09</td>
</tr>
<tr>
<td>General average</td>
<td>13.70</td>
<td>4.00</td>
<td>6.37</td>
</tr>
</tbody>
</table>

Table 2. Average computational times (s) for the three formulations.
# 5. Computational experiments

## Results

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>FB</th>
<th>GA</th>
<th>GD</th>
<th>SM1</th>
<th>SM2</th>
<th>PB1</th>
<th>PB2</th>
<th>PH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance size</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 × 13</td>
<td>0.043</td>
<td>2.163</td>
<td>1.930</td>
<td>2.807</td>
<td>5.714</td>
<td>5.251</td>
<td>2.481</td>
<td>0.001</td>
</tr>
<tr>
<td>10 × 26</td>
<td>0.017</td>
<td>1.813</td>
<td>3.334</td>
<td>4.206</td>
<td>6.663</td>
<td>4.977</td>
<td>2.185</td>
<td>0.007</td>
</tr>
<tr>
<td>20 × 13</td>
<td>0.035</td>
<td>2.249</td>
<td>2.651</td>
<td>2.641</td>
<td>5.830</td>
<td>4.506</td>
<td>2.221</td>
<td>0.030</td>
</tr>
<tr>
<td>20 × 26</td>
<td>0.021</td>
<td>2.024</td>
<td>3.702</td>
<td>2.670</td>
<td>5.188</td>
<td>3.431</td>
<td>2.825</td>
<td>0.019</td>
</tr>
<tr>
<td>Subgroup</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1</td>
<td>0.046</td>
<td>0.830</td>
<td>2.291</td>
<td>2.548</td>
<td>5.712</td>
<td>5.514</td>
<td>2.455</td>
<td>0.028</td>
</tr>
<tr>
<td>S2</td>
<td>0.023</td>
<td>3.212</td>
<td>3.225</td>
<td>4.023</td>
<td>5.985</td>
<td>3.738</td>
<td>2.514</td>
<td>0.012</td>
</tr>
<tr>
<td>S3</td>
<td>0.021</td>
<td>0.826</td>
<td>2.428</td>
<td>2.024</td>
<td>5.359</td>
<td>5.255</td>
<td>2.305</td>
<td>0.013</td>
</tr>
<tr>
<td>S4</td>
<td>0.017</td>
<td>3.453</td>
<td>3.349</td>
<td>3.657</td>
<td>5.545</td>
<td>3.654</td>
<td>2.456</td>
<td>0.011</td>
</tr>
<tr>
<td>S5</td>
<td>0.039</td>
<td>0.828</td>
<td>2.377</td>
<td>2.135</td>
<td>5.695</td>
<td>5.338</td>
<td>2.242</td>
<td>0.011</td>
</tr>
<tr>
<td>S6</td>
<td>0.028</td>
<td>3.223</td>
<td>3.756</td>
<td>4.099</td>
<td>6.795</td>
<td>3.749</td>
<td>2.596</td>
<td>0.010</td>
</tr>
<tr>
<td>Global results</td>
<td>0.029</td>
<td>2.062</td>
<td>2.904</td>
<td>3.081</td>
<td>5.848</td>
<td>4.541</td>
<td>2.428</td>
<td>0.014</td>
</tr>
<tr>
<td>Average (%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimum (%)</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Maximum (%)</td>
<td>1.876</td>
<td>9.229</td>
<td>9.556</td>
<td>18.100</td>
<td>20.000</td>
<td>14.718</td>
<td>9.097</td>
<td>0.778</td>
</tr>
<tr>
<td>Number of optimal solutions</td>
<td>619</td>
<td>138</td>
<td>41</td>
<td>119</td>
<td>37</td>
<td>26</td>
<td>114</td>
<td>646</td>
</tr>
</tbody>
</table>

Table 3. Average deviation with respect to the optimum for tested heuristics.
5. Conclusions

Overview

- Joint replenishment problem
- New formulations
- New heuristic

Further research

- Capacity restrictions