General lotsizing problem in a closed-loop supply chain with uncertain returns

Guillaume Amand, Yasemin Arda

February 7, 2013
1 Introduction

2 Deterministic Model

3 Stochastic Model

4 Approximate dynamic programming

5 Future work
Introduction

- Uncertainty in the return process is a common feature of closed loop supply chains.
- The uncertain quantity of returned items affects the production process.

Aim

Develop a mathematical model and an efficient algorithm to solve a general lotsizing and scheduling problem with uncertain returns.
Definition: Markov decision process

A Markov decision process is a 4-uple \((S, A, P(., .), V(., .))\) where:

- \(S\) is a finite set of states.
- \(A\) is a finite set of actions.
- \(P_a(s, s')\) is the probability that action \(a\) in state \(s\) will lead to state \(s'\).
- \(V(s, s')\) is the immediate reward received after transition from state \(s\) to state \(s'\).
There is a single production line without work-in-process inventories.

This line produces several products in lots. The size of each lot may vary and each product has a given production rate.

The production planning is realized for several periods.

The production capacity is limited but may vary between periods.
The demand for each product is considered as deterministic over the planning horizon.

Backorders are not allowed.

Building up an inventory is possible for the returned products and for the finished products.
Setup features

There are setup costs and times incurred whenever the production is switched from one product to another. The setup time consumes the capacity of the production line.

Setup costs and times are sequence dependent, i.e., they are determined based on the product produced before the changeover and the product produced after the changeover.
Production uses either returned items or new items.
The quantity of returned items that will be available in each period is not known with certainty.
The amount of available new items is considered unlimited.
Costs features

- Both returned items and end-products incur inventory costs.
- There is a cost per new item used. On the other hand, using returned items is free.
- Sequence dependant setup costs are considered.
Problem

Find a production schedule that minimizes the expected cost over the planning horizon, respects the production capacity, and satisfies demand at each period.
1. Introduction

2. Deterministic Model

3. Stochastic Model

4. Approximate dynamic programming

5. Future work
Deterministic Model

- One way to deal with the scheduling aspect is to divide each period into sub-periods.
- Only one type of item can be produced during a sub-period.
- This is the most common approach used in the literature (Mohammadi et al. (2010), Clark and Clark (2000), Fleischmann and Meyer (1997), Araujo et al. (2007)).
Deterministic Model

Min

\[
\sum_{n=1}^{N} \sum_{o=1}^{P} \sum_{p=1}^{P} CS_{op} \cdot y_{opn}
\]

\[
+ \sum_{t=1}^{T} \sum_{p=1}^{P} CJ_{p} \cdot w_{pt} + CB_{p} \cdot x_{pt}^{n} + CI_{p} \cdot z_{pt}
\]

s.t

\[
x_{pt}^{r} + x_{pt}^{n} \leq \sum_{o=1}^{P} \sum_{n=F_t}^{L_t} y_{opn} \cdot C_t \quad \forall p, t
\]

\[
z_{pt} = z_{p,t-1} + R_{pt} - x_{pt}^{r} \quad \forall p, t
\]

\[
w_{pt} = w_{p,t-1} + x_{pt}^{r} + x_{pt}^{n} - D_{pt} \quad \forall p, t
\]

\[
\sum_{p=1}^{P} L_{p} \cdot (x_{pt}^{r} + x_{pt}^{n}) + \sum_{n=F_t}^{L_t} \sum_{o=1}^{P} \sum_{p=1}^{P} S_{op} \cdot y_{opn} \leq C_t \quad \forall t
\]
Deterministic Model

\[ \sum_{p=1}^{P} y_{O_0p1} = 1 \]

\[ \sum_{p=1}^{P} y_{op1} = 0 \quad \forall o \neq O_0 \]

\[ \sum_{o=1}^{N} y_{opn} = \sum_{q=1}^{N} y_{pq,n+1} \quad \forall p, n \neq N \]

\[ y_{opn} \in \{0; 1\} \quad \forall o, p, n \]

\[ x^r_{pt}, x^n_{pt}, z_{pt}, w_{pt}, b_{pt} \geq 0 \quad \forall p, t \]
Introduction

Deterministic Model

Stochastic Model

Approximate dynamic programming

Future work
Stochastic Model

During a period, the following sequence of events occurs:

1. Decisions are made about production and inventories.
2. Returns become available.
3. Production starts and demand is satisfied.
Min $CJ \cdot w_1 + CS \cdot y_1 + E_{R_1} \left[ f_1(S_1, R_1) \right]$

s.t $x_1 \leq C_1 \cdot y_1$

$w_1 = w_0 + x_1 - D_1$

$x_1, w_1 \geq 0$

$y_1 \in \{0, 1\}$
where \( f_t(S_t, R_t) \) is equal to:

\[
\begin{align*}
\text{Min} & \quad Cl \cdot z_t + CB \cdot x_t^n + CJ \cdot w_{t+1} \\
& \quad + CS \cdot y_{t+1} + E_{R_{t+1}} \left[ f_{t+1}(X_{t+1}, R_{t+1}) \right] \\
\text{s.t} & \quad x_t^n + x_t^r = x_t \\
& \quad z_t = z_{t-1} + R_t - x_t^r \\
& \quad x_{t+1} \leq C_{t+1} \cdot y_{t+1} \\
& \quad w_{t+1} = w_t + x_{t+1} - D_{t+1} \\
& \quad x_t^n, x_t^r, x_{t+1}, w_{t+1} \geq 0 \\
& \quad y_{t+1} \in \{0, 1\}
\end{align*}
\]
Markov decision process representation

- A state is a triple \((t, z_{t-1}, w_t)\).
- \(y_t\) and \(x_t\) define the set of actions.
- The transition function is defined by the constraints.
- The reward perceived after a transition is given by the objective function.

The transition

\[
(t - 1, Z_{t-2}, W_{t-1}) \xrightarrow{(y_t, x_t, R_t)} (t, Z_{t-1}, W_t)
\]

where:

\[
Z_{t-1} = \max(0, Z_{t-2} + R_t - x_{t-1})
\]
\[
W_t = W_{t-1} + x_t - D_t
\]
1 Introduction

2 Deterministic Model

3 Stochastic Model

4 Approximate dynamic programming

5 Future work
**Approximate Dynamic Programming**

This technique is described in Powell (2011) and has been used in various production problems (Qiu and Loulou (1995), Erdelyi and Topaloglu (2011)).

**Idea**

Iteratively solve an approximation of the deterministic problem. After each iteration, use the preceding results to affine the approximation.
At each iteration of the algorithm, the following sequence of operations are:

1. Select a scenario \((R_1, \ldots, R_T)\).
2. Solve the sub-problem \(f_t(X_t, R_t)\) for each period where the expectation is replaced by an approximation.
3. Update the approximation using the obtained results.
The current characteristics of the algorithm are:

- The algorithm stops after a certain number of iterations.
- The scenario selection is totally random.
- The transition cost function is represented as a table.
- Update of the transition cost table uses a k-nearest neighbour procedure.
## Preliminary Results

<table>
<thead>
<tr>
<th>$T$</th>
<th>5</th>
<th>10</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu$</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>ACE</td>
<td>3%</td>
<td>3%</td>
<td>2%</td>
</tr>
<tr>
<td>HCE</td>
<td>7%</td>
<td>5%</td>
<td>7%</td>
</tr>
<tr>
<td>Time (s)</td>
<td>65</td>
<td>117</td>
<td>181</td>
</tr>
</tbody>
</table>
Future work

- Use other types of procedures to improve the algorithm.
- Expand the algorithm to the multi-product problem.
Future work

Thank you for your attention!


