Lectures – Dynamic Pricing

Part I – Background Material

BRNO – 2009

By

Kjetil K. Haugen
The classic linear Monopoly model

- Assumptions: (linear demand and linear and proportional costs)
  \[ c(Q) = cQ \]
  \[ P = a - bQ \]
- \( a, b > 0 \) \( \Rightarrow \) Normal good
- Profit function:
  \[ \Pi(Q) = R(Q) - C(Q) \]
  \[ = PQ - cQ \]
  \[ = (a - bQ)Q - cQ \]
Monopoly solution

- **Analytically:** (Max $\Pi$)
  \[
  \Pi'(Q) = 0 \\
  \Rightarrow (a - c) - 2bQ^* = 0 \\
  \Rightarrow Q^* = \frac{a - c}{2b}
  \]

- **Graphically:**
  \[
  \Pi(Q) = R(Q) - c(Q) \\
  \Pi'(Q) = 0 \\
  \Rightarrow MR(Q) - MC(Q) = 0 \\
  \Rightarrow MR = MC
  \]

- **MR and MC:**
  \[
  R(Q) = P(Q)Q \\
  = (a - bQ)Q = aQ - bQ^2 \\
  \Rightarrow MR = R'(Q) = a - 2bQ \\
  MC = c'(Q) = c
  \]
Perfect Competition

- **Monopoly:**
  - 1 producer
  - $\infty$ consumers

- **Perfect competition**
  - $\infty$ producers
  - $\infty$ consumers
  - $\Delta Q$ give no $\Delta P$

\[ \Pi(Q) = P \cdot Q - cQ \]
\[ \Pi'(Q) = P - MC = 0 \]
\[ \Rightarrow P = MC \]
A practical example – Dynamic Monopoly (1)

• Look at a grocery store selling milk.
• Shop owner has experienced a customer daily demand pattern like:
A practical example – Dynamic Monopoly (2)

• Suppose that in order to serve such a daily demand, the shop needs 4 employees for the tops, 1 for the bottoms and 2 for the average demand.

• Could you device a simple dynamic pricing strategy for improving profits compared to a fixed price strategy?

• What determines if a dynamic pricing strategy is better than a fixed price strategy?

• What about existence of other milk sellers?
A simple lot-size Dynamic Pricing Problem (Dynamic Monopoly)

• Assume:
  – A producer produce **one** product in $T$ timeperiods
  – In each of these $T$ timeperiods, a demand function $\phi_t(p_t)$ exist. These functions are given and represent consumer (steady) preferences.
  – The producer can store finished products and the products are assumed non-perishable
  – Product storage leads to inventory costs - $h_t$, also production costs - $c_t$ are present. We assume linearity and proportionality for these costs.
  – Production capacity is limited by $R_t$
A Mathematical Programming formulation (NLP)

\[
Max \ Z = \sum_{t=1}^{T} \phi_t(p_t)p_t - c_t x_t - h_t I_t
\]

s.t

\[
I_t = I_{t-1} + x_t - \phi_t(p_t) \ \forall t
\]

\[
x_t \leq R_t \ \forall t
\]

\[
x_t, I_t, p_t \geq 0
\]

\(X_t\): Amount produced in period \(t\)

\(I_t\): Amount stored between periods \(t-1, t\)

\(p_t\): Price set in period \(t\)
Comparing various versions of this model

- **Assumptions:**
  - Linear demand: \( \phi_t(p_t) = a_t - b_t p_t \)

- **Case 1:**
  - Fixed given price (LP – cost minimization)

- **Case 2:**
  - Constant variable price (QP – profit maximisation)

- **Case 3:**
  - Discrete price choice (MILP – profit maximization)

- **Case 4:**
  - Dynamic continuous price (QP – profit maximisation)
A Mathematical Programming formulation (Case 1)

\[
\text{Min } Z = \sum_{t=1}^{T} c_t x_t + h_t I_t \\
\text{s.t.} \\
I_t = I_{t-1} + x_t - d_t \ \forall t \\
x_t \leq R_t \ \forall t \\
x_t, I_t \geq 0
\]

A given constant price \( p_t = p^0 \ \forall t \) leads to a constant term in the objective. Maximising a constant minus ”something” is the same as minimising ”something”. Then defining \( \phi_t(p^0) = d_t \) yields the above LP-model.
A Mathematical Programming formulation (Case 2)

\[
\text{Max } Z = Ap - Bp^2 \quad - \quad \left\{ \sum_{t=1}^{T} c_t x_t + h_t I_t \right\}
\]

s.t

\[
I_t = I_{t-1} + x_t - \left\{ a_t - b_t p \right\} \forall t
\]

\[
x_t \leq R_t \forall t
\]

\[
x_t, I_t, p \geq 0
\]

if \( p_t = p \forall t \) then

\[
\sum_{t=1}^{T} (a_t - b_t p) p = p \sum_{t=1}^{T} a_t - p^2 \sum_{t=1}^{T} b_t = Ap - Bp^2
\]
A Mathematical Programming formulation (Case 3)

Discrete prices implies a given set of possible prices in all periods to choose from.

\[ p_{it} : \text{Price alternative } i, i \in \{1, \ldots I\} \text{ in period } t \]

\[ \gamma_{it} = \begin{cases} 1 & \text{price } i \text{ is chosen in period } t \\ 0 & \text{otherwise} \end{cases} \]

\[ d_{it} = a_t - b_t p_{it} \]

\[ \theta_{it} = d_{it} p_{it} \]

then the revenue can be expressed by:

\[ \sum_{i=1}^{I} \theta_{it} \gamma_{it} \quad \forall t \]

and to secure only one picked price:

\[ \sum_{i=1}^{I} \gamma_{it} = 1 \quad \forall t \]
A Mathematical Programming formulation (Case 3)

\[
\text{Max } Z = \sum_{t=1}^{T} \left\{ \sum_{i=1}^{I} \theta_{it} \gamma_{it} \right\} - c_t x_t - h_t I_t
\]

s.t

\[
I_t = I_{t-1} + x_t - \sum_{i=1}^{I} d_{it} \gamma_{it} \quad \forall t
\]

\[
\sum_{i=1}^{I} \gamma_{it} = 1 \quad \forall t
\]

\[
x_t \leq R_t \quad \forall t
\]

\[
x_t, I_t \geq 0
\]

\[
\gamma_{it} \in \{1,0\}
\]
A Mathematical Programming formulation (Case 4)

\[
Max \ Z = \sum_{t=1}^{T} (a_t - b_t p_t) p_t - c_t x_t - h_t I_t \\
\text{s.t.} \\
I_t = I_{t-1} + x_t - (a_t - b_t p_t) \ \forall t \\
x_t \leq R_t \ \forall t \\
x_t , I_t , p_t \geq 0
\]
Solving some cases:

- **Case data:**
  - $T=4$
  - $R_t=20 \quad \forall \ t$
  - $c_t=20 \quad \forall \ t$
  - $h_t=2 \quad \forall \ t$
  - Demand functions (see individual case presentations)
Case 1 (Cost minimization)

- Only one given price ($p^* = 95$)
- Constant capacity constraint ($R_t = 20 \forall t$)

$$\Pi^* = 4294,5$$
Case 2: (max $\Pi$ – single price)

- Only one (but variable) price ($p^* = 88$)
- Constant capacity constraint ($R_t = 20 \ \forall t$)

$\Pi^* = 5658$
Case 3: (Max $\Pi$ discrete prices - 3 options 90, 80, 70)

- Optimal discrete prices - $p^*=[90, 70, 80, 80]$
- Constant capacity constraint ($R_t = 20 \ \forall t$)

$\Pi^* = 5973.33$
Case 4: (Max $\Pi$ - full continuous dynamic pricing)

- Optimal prices - $p^*=[88, 66.29, 77.79, 92.74]$
- Constant capacity constraint ($R_t = 20 \ \forall t$)

$\Pi^* = 6154.33$
# Summing up

<table>
<thead>
<tr>
<th></th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Pi^*$</td>
<td>4294.5</td>
<td>5658</td>
<td>5973.33</td>
<td>6154.33</td>
</tr>
<tr>
<td>Δ%</td>
<td>-</td>
<td>31.75%</td>
<td>5.57%</td>
<td>3.03%</td>
</tr>
<tr>
<td>Δ% - Total</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>43.31%</td>
</tr>
</tbody>
</table>

![Bar chart showing case comparisons](chart.png)

- **Case 1**: 4294.5
- **Case 2**: 5658
- **Case 3**: 5973.33
- **Case 4**: 6154.33

The Δ% values indicate the percentage change for each case compared to a baseline.

- **Δ%**: 31.75% for Case 2, 5.57% for Case 3, and 3.03% for Case 4.

- **Δ% - Total**: 43.31% for all cases combined.
Some discussion topics

• How does the effects of Dynamic pricing relate to:
  – Demand time variability?
  – Size of storage costs?
  – Size of Capacity Constraints?
  – What risks will a manufacturer face by applying Dynamic Pricing in a non-monopolistic (oligopolistic) environment?
  – Does Dynamic Pricing in a Perfectly Competitive market make sense?

Use the DPD (Dynamic Pricing Demonstrator) to discuss these questions!
Some extensions:

• Multiple products
• Set-up costs (and times)
• Marketing
• Uncertainty
• Gaming (competition among Dynamic Pricer’s)
Lectures – Dynamic Pricing
Part II – Research Material

BRNO – 2008

By

Kjetil K. Haugen
The profit maximizing capacititated lot-size (PCLSP) problem

by Kjetil K. Haugen
Asmund Olstad
and
Bård I. Pettersen

Molde University College
Servicebox 8, N-6405 Molde, Norway

E-mail: Kjetil.Haugen@hiMolde.no

Idea – Abstract

- Introduce a "new" set of lot-sizing models including pricing – PCLSP.

- PCLSP – practically at least as relevant as CLSP.

- PCLSP – computationally more feasible than CLSP.
Outline

1) Introduce LSP, CLSP and PCLSP.

2) Brief discussion of algorithmic properties of LSP and CLSP.

3) Introduce a Lagrange Relaxation algorithm and discuss algorithmic performance by example.

4) Introduce new algorithms and compare with CLSP.

5) Discuss practical relevance of PCLSP.
The simple lot-size problem (LSP) (1)

\[ \text{Min } Z = \sum_{t=1}^{T} s_t \delta_t + h_t I_t + c_t x_t \]  \hspace{1cm} (1) \\

s.t.

\[ x_t + I_{t-1} - I_t = d_t \quad \forall t \]  \hspace{1cm} (2) \\
\[ 0 \leq x_t \leq M_t \delta_t \quad \forall t \]  \hspace{1cm} (3) \\
\[ I_t \geq 0, \quad \forall t \]  \hspace{1cm} (4) \\
\[ \delta_t \in \{0,1\} \quad \forall t \]  \hspace{1cm} (5)
The simple lot-size problem (LSP) (2)

Decision variables:

\[ x_t = \text{amount produced in period } t \]
\[ I_t = \text{inventory between } t, t+1 \]
\[ \delta_t = 1 \text{ if } x_t > 0 \text{ in period } t; 0 \text{ otherwise} \]

Parameters:

\[ T = \text{number of time periods} \]
\[ s_t = \text{setup cost in period } t \]
\[ h_t = \text{storage cost between } t, t+1 \]
\[ c_t = \text{unit production cost in period } t \]
\[ M_t = \text{"Big M" in period } t \]
The simple lot-size problem (LSP) (3)

- Problem characteristics

Basic trade off:

- Many set-ups ⇒ $\sum_t s_t \delta_t \uparrow$ and $\sum_t h_t I_t \downarrow$

- Few set-ups ⇒ $\sum_t s_t \delta_t \downarrow$ and $\sum_t h_t I_t \uparrow$

Slight generalization of EOQ-model:

Given $d_t = d, c_t = c, h_t = h \forall t \Rightarrow LSP = EOQ$
The simple lot-size problem (LSP) (4)

- Algorithmic characteristics

- Polynomial DP-algorithm by Wagner and Whitin (1950).

- Solves very fast.

- Planning horizon theorems; $x_t \cdot I_t = 0$.

- Interesting candidate as sub-problem solver in more advanced lot-size problems.
The capacitated problem (CLSP) (1)

\[
\text{Min } Z = \sum_{t=1}^{T} \sum_{j=1}^{J} s_{jt} \delta_{jt} + h_{jt} I_{jt} + c_{jt} x_{jt} \tag{6}
\]

s.t.

\[
\sum_{j=1}^{J} a_{jt} x_{jt} \leq R_t \quad \forall t \tag{7}
\]

\[
x_{jt} + I_{j,t-1} - I_{jt} = d_{jt} \quad \forall jt \tag{8}
\]

\[
0 \leq x_{jt} \leq M_{jt} \delta_{jt} \quad \forall jt \tag{9}
\]

\[
I_{jt} \geq 0, \quad \forall jt \tag{10}
\]

\[
\delta_{jt} \in \{0, 1\} \quad \forall jt \tag{11}
\]
CLSP (2)

Decision variables:

\[ x_{jt} = \text{amount of item } j \text{ produced in } t \]
\[ I_{jt} = \text{inventory of item } j \text{ between } t, t+1 \]
\[ \delta_{jt} = \begin{cases} 1 & \text{if item } j \text{ is produced in period } t \\ 0 & \text{otherwise} \end{cases} \]

Parameters:

\[ T = \text{number of time periods} \]
\[ J = \text{number of items} \]
\[ s_{jt} = \text{setup cost for item } j \text{ in period } t \]
\[ h_{jt} = \text{storage cost, item } j \text{ between } t, t+1 \]
\[ c_{jt} = \text{unit production cost, item } j \text{ at } t \]
\[ a_{jt} = \text{resource used, item } j \text{ at } t \]
\[ R_t = \text{capacity resource available at } t \]
\[ M_{jt} = \sum_{s=t}^{T} d_{js} \]
CLSP (2) – characteristics

- Much harder to solve compared to LSP.

- Reason:
  
  Violation of capacity constraint $\Rightarrow$ ”moving production around” (combinatorial).

- Due to non existence of polynomial algorithms (NP-hardness) heuristical (Lagrange relaxation based) approaches common.

- A very ”popular” OR research problem.

- Still: typical problem sizes not much larger than $100 \times 100$ – not satisfactory given product variety today.
Max \[ Z = \sum_{t=1}^{T} \sum_{j=1}^{J} \left[ d_{jt}p_{jt} - s_{jt}\delta_{jt} - h_{jt}I_{jt} - c_{jt}x_{jt} \right] \]

s.t.

\[ \alpha_{jt} - \beta_{jt} \cdot p_{jt} = d_{jt} \quad \forall j \quad \forall t \quad (13) \]

\[ \sum_{j=1}^{J} a_{jt}x_{jt} \leq R_{t} \quad \forall t \quad (14) \]

\[ x_{jt} + I_{j,t-1} - I_{jt} = d_{jt} \quad \forall j \quad \forall t \quad (15) \]

\[ 0 \leq x_{jt} \leq M_{jt}\delta_{jt} \quad \forall j \quad \forall t \quad (16) \]

\[ I_{jt} \geq 0, \quad \forall j \quad \forall t \quad (17) \]

\[ \delta_{jt} \in \{0, 1\} \quad \forall j \quad \forall t \quad (18) \]

\[ \frac{\alpha_{jt}}{\beta_{jt}} \geq p_{jt} \geq 0 \quad \forall j \quad \forall t \quad (19) \]
PCLSP (2)

Decision variables added to CLSP:

\[ p_{jt} = \text{price of item } j \text{ in period } t \]

Parameters added to CLSP:

\[ \alpha_{jt} = \text{constant in linear demand, item } j \text{ at } t \]
\[ \beta_{jt} = \text{slope in linear demand, item } j \text{ at } t \]
PCLSP characteristics

- Linear demand, Monopoly assumption (unrealistic).

- PCLSP is a generalization of CLSP. *

- Immediate feasible solutions are obtainable (as opposed to CLSP) by "pricing out". †

- PLSP (uncapacitated single item version) is well known from OR-literature. Thomas (1970) constructed a polynomial DP-algorithm with complexity as of the Wagner/Whitin algorithm.

*Easy to see by the special case \( p_{jt} = \hat{p}_{jt} \) where \( \hat{p}_{jt} \) are assumed constant.

†That is, any capacity constraint violation can be "removed" as any demand may be forced to zero \( (p_{jt} = \frac{\alpha_j}{\beta_j}) \).
Basic hypothesis

The added problem flexibility obtained by introduction of price variables should make the problem easier to solve.
LUBP sub problems

By relaxing the capacity constraint (14), the PCLSP-problem (12) – (19) may be expressed as:

$$\text{Max } Z = Z + \sum_{t=1}^{T} \lambda_t \left( R_t - \sum_{j=1}^{J} a_{jt} x_{jt} \right)$$

s.t. constraints (13) to (19)

It is "straightforward" to adjust the DP-algorithm by Thomas (1970) to provide very efficient solutions to the LUBP sub-problems.

Note: Solving LUBP yields upper bound on PCLSP.
LLBP sub problems

If $\delta_{ij}$’s are fixed (Set-up structure) in PCLSP, a standard quadratic programming problem is obtained:

$$\text{Max } Z = \sum_{t=1}^{T} \sum_{j=1}^{J} \left[ d_{jt}p_{jt} - h_{jt}I_{jt} - c_{jt}x_{jt} \right]$$

s.t. \hspace{1cm} (13), (15), (17), (19).

Any solution to LLBP is feasible and typically non-optimal. Hence, Any solution to LLBP is a lower bound to PCLSP.
Algorithmic structure

LUBP

Set-up structure $Z_{PCLSP}$ $\lambda_t^k$

LLBP
Algorithm: Lagrange relaxation

0. Define $\lambda_t = 0, \forall t = 1, 2, \ldots, T$

1. Solve LUBP (Obtain Set-up structure)

2. Solve LLBP (for Set-up structure obtained in step 1.) Define $\lambda_t$ from this solution as $\lambda^k_t$, where $k$ denotes iteration count.

   Stop if: (define reasonable stopping criteria)

3. Update $\lambda^k_t$ by smoothing: $\lambda^k_{t+1} = \theta_k \cdot \lambda^k_t + (1-\theta_k)\lambda^{k-1}_t, \forall t$, where $\theta_k$ is a smoothing parameter, $0 \leq \theta_k \leq 1$

4. Go to step 1.
Some results (1)

<table>
<thead>
<tr>
<th>Problem</th>
<th>CLSP</th>
<th>PCLSP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#It.</td>
<td>G(%)</td>
</tr>
<tr>
<td>CAP93%</td>
<td>50</td>
<td>10</td>
</tr>
<tr>
<td>CAP91%</td>
<td>50</td>
<td>3</td>
</tr>
<tr>
<td>CAP73%</td>
<td>50</td>
<td>2</td>
</tr>
<tr>
<td>CAP61%</td>
<td>50</td>
<td>1</td>
</tr>
</tbody>
</table>

- Cases captured from Thizy and Wassenhove (1985). 8 products over 8 time periods.

- #It. means number of iterations performed in algorithm.

- G(%) means gap in percent or \( \frac{Z_{LUBP}-Z_{LLBP}}{Z_{LUBP}} \times 100 \).
Some results (2)

<table>
<thead>
<tr>
<th>Problem</th>
<th>CLSP</th>
<th>PCLSP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#It.</td>
<td>G(%)</td>
</tr>
<tr>
<td>CAP93%</td>
<td>50</td>
<td>10</td>
</tr>
<tr>
<td>CAP91%</td>
<td>50</td>
<td>3</td>
</tr>
<tr>
<td>CAP73%</td>
<td>50</td>
<td>2</td>
</tr>
<tr>
<td>CAP61%</td>
<td>50</td>
<td>1</td>
</tr>
</tbody>
</table>

The same cases with Gaps after one iteration in PCLSP compared to 50 iterations in CLSP.
PCLSP – practical relevance

Demonstrated (by some examples) that PCLSP solves significantly faster than CLSP.

What about practical relevance?

Obvious facts:

- If monopolistic market conditions ⇒ PCLSP > CLSP

- If Free markets (price taking behavior) CLSP is "correct"

- Most markets are neither (oligopoly) – what then?
Price constraints


- Given "relatively small" price changes, underlying Nash equilibrium may be stable.

- Price constraints may serve the purpose

- Relatively simple to introduce (no radical algorithmic changes)