Lecture outline

- Basic inventory problems
- The economic order quantity
- An inventory game
- Multiperiod lot sizing
  » Math programming formulation
  » Heuristics
  » Wagner-Whitin algorithm
Basic inventory problems

Examples:

» Products:
  • Customers consume products over time.
  • Store replenishes periodically.

» People with specialized training:
  • People randomly leave the company over time.
  • Company periodically hires new graduates.

» Water (management of dams).
  • Rainfall randomly replenishes reservoirs.
  • Release water from dam to maintain level.

» Oil being stored in storage tanks:
  • Oil is steadily consumed.
  • Periodically is replenished from tankers.
Basic inventory problems

Examples:

» Housing stock
  • Houses are continually being purchased.
  • Developers produce new developments or apartment buildings.

» Financial resources (startup company)
  • Cash is used to build up the company.
  • Periodically fresh capital is raised from venture capitalists.

» Features in a software program:
  • Accumulating features in a software program in response to user requests and the ideas of developers, or due to bug fixes.
  • Periodically ship a new version of the program.

» Purchasing stock:
  • Funds become available for investment.
  • Periodically purchase new shares of stock.
# Basic inventory problems

<table>
<thead>
<tr>
<th>Statement Date</th>
<th>Account Number</th>
<th>Account Summary for the Period</th>
<th>Replenishment Amount</th>
<th>Replenishment Method</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Tag</th>
<th>Deposit</th>
<th>Beginning Balance</th>
<th>Tolls &amp; Fees</th>
<th>Payments &amp; Credits</th>
<th>Ending Balance</th>
<th>Replenishment Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.00</td>
<td>22.07</td>
<td>27.25</td>
<td>25.00</td>
<td>19.82</td>
<td>10.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date/Time</th>
<th>Tag</th>
<th>Transaction</th>
<th>Entry Plaza Lane</th>
<th>Exit Plaza Lane</th>
<th>Class</th>
<th>Amount</th>
<th>Balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>11/03 14:20</td>
<td>02200917779</td>
<td>New Jersey Turnpike Toll</td>
<td>9</td>
<td>05E</td>
<td>13A</td>
<td>13X</td>
<td>-1.55</td>
</tr>
<tr>
<td>11/06 22:21</td>
<td>02200917779</td>
<td>New Jersey Turnpike Toll</td>
<td>13A</td>
<td>05E</td>
<td>9</td>
<td>13X</td>
<td>-1.45</td>
</tr>
<tr>
<td>11/11 15:09</td>
<td>02200917779</td>
<td>New York State Thruway Toll</td>
<td>15</td>
<td>05E</td>
<td>24</td>
<td>08S</td>
<td>-3.65</td>
</tr>
<tr>
<td>11/13 14:27</td>
<td>02200917779</td>
<td>New York State Thruway Toll</td>
<td>24</td>
<td>04E</td>
<td>15</td>
<td>08W</td>
<td>-3.65</td>
</tr>
<tr>
<td>11/13 17:35</td>
<td>02200917779</td>
<td>New Jersey Turnpike Toll</td>
<td>10</td>
<td>11E</td>
<td>7A</td>
<td>06X</td>
<td>-0.75</td>
</tr>
<tr>
<td>12/15 16:59</td>
<td>02200917779</td>
<td>New Jersey Turnpike Toll</td>
<td>9</td>
<td>05E</td>
<td>16E</td>
<td>01X</td>
<td>-2.30</td>
</tr>
<tr>
<td>12/15 17:42</td>
<td>02200917779</td>
<td>PANYNJ Toll</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12/16 10:53</td>
<td>02200917779</td>
<td>New Jersey Turnpike Toll</td>
<td>14C</td>
<td>08E</td>
<td>9</td>
<td>12X</td>
<td>-2.40</td>
</tr>
<tr>
<td>12/17 16:50</td>
<td>02200917779</td>
<td>New Jersey Turnpike Toll</td>
<td>9</td>
<td>05E</td>
<td>13A</td>
<td>13X</td>
<td>-1.55</td>
</tr>
<tr>
<td>12/18 02:35</td>
<td></td>
<td>Replenishment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12/19 20:53</td>
<td>02200917779</td>
<td>New Jersey Turnpike Toll</td>
<td>13A</td>
<td>06E</td>
<td>9</td>
<td>13X</td>
<td>-1.45</td>
</tr>
<tr>
<td>12/26 13:44</td>
<td>02200917779</td>
<td>New York State Thruway Toll</td>
<td>15</td>
<td>02E</td>
<td>23</td>
<td>06E</td>
<td>-3.50</td>
</tr>
</tbody>
</table>
Basic inventory problems

- Mutual fund cash balance

Stock market       Cash       Investor

How much cash do we keep on hand to strike a balance between the deposits and withdrawals of investors, and the behavior of the market?
Basic inventory problems

There are a number of ways to refer to storing resources for the future:

- Physical resources
  - Inventory
  - Stockpile
  - Stock

- Financial resources
  - Savings
  - Nest egg
  - Reserve
Basic inventory problems

- Reasons for holding inventories:
  » Economies of scale
    • Batches of goods
    • Discounts (purchasing)
    • Transportation economies (e.g. shipping in bulk)
  » Uncertainties
    • Demand
    • Order lead times
    • Supply/price of raw materials (OPEC)
    • Supply/price of components (strikes)
    • Quality control
  » Speculation
    • Commodities prices
    • Currency fluctuations
Basic inventory problems

- Reasons for holding inventories
  - Transportation
    - In-transit or pipeline inventories
  - Smoothing production
    - Respond to seasonal patterns in demand
    - Seasonal production of some items
      - Certain foods
      - Syrup
      - Snow
      - Students
  - Control costs
    - Lower inventories requires more sophisticated control systems
Basic inventory problems

The lot sizing problem

» Often, there are economies of scale when ordering new resources:
  • Raising operating capital
    – There is a fixed cost to going to the capital markets
    – Just as much work to raise $1m as $5m
  • Shipping the latest version of a software program
    – New features are added over time
    – There is a fixed cost of shipping a new version of the code
    – How many new features do you add before you ship the code?
  • Ordering new product for a store shelf
    – Fixed cost for placing and shipping an order
Basic inventory problems

- An aging and replenishment process (negative drift):
  » State is inventory – drift is due to customer demand.
### Basic inventory problems

- **Indexing time:**
  - Deterministic indexing – Index based on when something happens.
  - Stochastic indexing – Index based on when something becomes known.
  - Deterministic indexing:

<table>
<thead>
<tr>
<th>Continuous time</th>
<th>$D_0$</th>
<th>$D_1$</th>
<th>$D_2$</th>
<th>$D_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t = 0$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$t = 1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$t = 2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$t = 3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Discrete time</th>
<th>$R_0, x_0$</th>
<th>$R_1, x_1$</th>
<th>$R_2, x_2$</th>
<th>$R_3, x_3$</th>
<th>$R_4, x_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t = 0$</td>
<td>$R_0, x_0$</td>
<td>$R_1, x_1$</td>
<td>$R_2, x_2$</td>
<td>$R_3, x_3$</td>
<td>$R_4, x_4$</td>
</tr>
</tbody>
</table>
Basic inventory problems

- Basic inventory equation
  » Deterministic indexing

\[ R_{t+1} = \max \{ 0, R_t + x_t - D_t \} = [R_t + x_t - D_t]^+ \]

- In deterministic indexing, everything is modeled at the beginning of a time period.
Basic inventory problems

- Stochastic indexing
  - Information arrives continuously over time
  - *A variable indexed by* $t$ *contains exogenous information up through time* $t$.

Continuous time

<table>
<thead>
<tr>
<th>$D_1$</th>
<th>$D_2$</th>
<th>$D_3$</th>
<th>$D_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t = 1$</td>
<td>$t = 2$</td>
<td>$t = 3$</td>
<td>$t = 4$</td>
</tr>
</tbody>
</table>

Discrete time

<table>
<thead>
<tr>
<th>$R_0, x_0$</th>
<th>$R_1, x_1$</th>
<th>$R_2, x_2$</th>
<th>$R_3, x_3$</th>
<th>$R_4, x_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t = 0$</td>
<td>$t = 1$</td>
<td>$t = 2$</td>
<td>$t = 3$</td>
<td>$t = 4$</td>
</tr>
</tbody>
</table>

- In stochastic indexing, everything is indexed at the end of a period.
Basic inventory problems

- Basic inventory equation
  - Stochastic indexing

\[ R_{t+1} = \max \{0, R_t + x_t - D_{t+1}\} = [R_t + x_t - D_{t+1}]^+ \]

Demand (random at time \( t \))

New orders

State (e.g. inventory level)
Basic inventory problems

- Inventory with positive drift
  » Cash is deposited in an account. Periodically the cash is invested in batch amounts to reduce transaction costs.
Basic inventory problems

- Basic inventory with positive drift
  - Blood donations, water reservoirs, ...

\[ R_{t+1} = \left[ R_t - x_t \right]^+ + D_{t+1} \]

- Exogenous drift
- Endogenously controlled shift
- State (e.g. accumulation)
Outline

- Basic inventory problems
- The economic order quantity
- An inventory game
- Multiperiod lot sizing
  - Math programming formulation
  - Heuristics
  - Wagner-Whitin algorithm
The economic order quantity

- Order costs
  - Fixed cost of placing an order
    - Paperwork, forms, telephone
    - Sending a truck out
    - Setting up a machine
    - Price structure imposed by supplier
  - Variable cost of ordering a certain amount of product
    - Variable cost may be fixed (a linear function).
    - Or there may be economies of placing larger orders.
The economic order quantity

Holding cost

» Storage (heat, electricity, supervision, etc.)
» Taxes and insurance
» Breakage, spoilage, deterioration and obsolescence
  • Careful with these - these “costs” convert to reduction in quantity.
» Opportunity cost (interest)
  • Hurdle rate for a company is generally much higher than bank interest rates.
  • Let $I =$ “interest rate”
  • $c_p =$ purchase cost of item
  • $c^h = Ic_p =$ holding cost (be careful with units; if $I$ is interest rate per year, $c^h$ is holding cost per year).
The economic order quantity

- Stockout cost
  » Cost of lost customers
  » Cost of pushing orders to future time periods

- Notes:
  » Stockout costs are not relevant in our simplest inventory system, because they cannot happen.
  » Stockouts arise when:
    • Demand is random
    • Demand varies over time with production capacities
    • Order costs may vary as a function of time, possibly exceeding the “benefit” of covering demand.
The economic order quantity

- Some assumptions:
  - Demand is deterministic with rate $\lambda$ per unit time.
  - Rate $\lambda$ is constant – stationary process.
  - Costs are stationary.
  - Orders arrive immediately.
  - All orders must be filled.
The economic order quantity

The basic inventory equation:

\[
R_{t+1} = [R_t + x_t - D_{t+1}]^+
\]

where:

- \( R_t \) = Inventory at start of time \( t \)
- \( x_t \) = Amount ordered at time \( t \)
- \( D_t \) = Demand during period \( t \)

Notation:

\([x]^+ = \max \{x, 0\}\)
The economic order quantity

**The cost function:**

\[ c(x_t, R_t) = \text{Total costs during period } t \text{ given order quantity } x_t \text{ and initial inventory } R_t \]

\[ = c^o(x_t) + c^h(R_t, x_t) \]

where:

- \( c^o(x_t) \) = Order costs
  \[ = \begin{cases} K + c^p x_t & x_t > 0 \\ 0 & x_t = 0 \end{cases} \]

- \( c^p \) = Unit purchasing costs

- \( c^h(R_t, x_t) \) = Holding costs
  \[ = c^h \int_0^{\Delta t} [R_t + x_t - \lambda z]^+ dz \]

- \( c^h \) = Unit holding costs per time
The economic order quantity

The cost function:

We can transform the nonlinear cost function into a linear one:

\[ c^o(x_t) = Ky_t + c^p x_t \]

where:

\[ x_t \geq 0 \]
\[ x_t \leq My_t \quad \text{M = "big M"} \]
\[ y_t \in (0,1) \]

If \( y_t = 0 \) then we force \( x_t = 0 \). Now we have transformed a nonlinear cost function into a linear one, but we have added an integer variable.
The economic order quantity

Infinite horizon problem:

We would like to solve:

$$\min \sum_{x_t, y_t}^{\infty} c(R_t, x_t, y_t)_{t=0}$$

This is a really big number! A more formal way to write it is as an average cost:

$$\lim_{T \to \infty} \frac{1}{T} \left\{ \min_{x_t, y_t} \sum_{t=0}^{T} c(R_t, x_t, y_t) \right\}$$
The economic order quantity

Intuition suggests that we let inventories drop to zero, and then “order up to” an amount $Q$:

$\tau = \text{reorder interval}$
The economic order quantity

Reformulate the problem in terms of cost per order interval:

\( Q \) = order quantity

\( \lambda \) = Demand rate per unit time (assumed constant and deterministic)

\( c^o (Q) \) = Order cost per order interval

\( = K \) (since we always make one order per interval).

\( \tau \) = Length of order interval

\( = \frac{Q}{\lambda} \)

\( c^p (Q) \) = Purchase costs per order interval

\( = c^p Q \)
The economic order quantity

\[ c^h(Q) = \text{Holding cost per order interval} \]
\[ = (\text{holding cost}) \cdot (\text{average inventory}) \cdot (\text{length of interval}) \]
\[ = c^h \left( \frac{Q}{2} \right) \tau \]
\[ = c^h \left( \frac{Q}{2} \right) \left( \frac{Q}{\lambda} \right) = c^h \left( \frac{Q^2}{2\lambda} \right) \]

\[ c(Q) = \text{Total cost per order interval} \]
\[ = c^o(Q) + c^p(Q) + c^h(Q) \]
\[ = K + c^p Q + c^h \left( \frac{Q^2}{2\lambda} \right) \]
The economic order quantity

How do we find the optimum value of Q?

We can try differentiating $C(Q)$ with respect to Q:

$$\frac{dC(Q)}{dQ} = 0 + c^p + 2c^h \frac{Q}{2\lambda} = 0$$

Solving for $Q$ gives us:

$$Q^* = -\frac{c^p \lambda}{c^h}$$

What went wrong?
The economic order quantity

Cost per cycle = K plus quantity proportional to green area
The economic order quantity
The economic order quantity

Q
The economic order quantity

The smaller $Q$ gets, the lower our costs per cycle.
The economic order quantity

Need to minimize cost per unit time, not cost per order interval. So, we want to solve:

\[
\min C^\tau (Q) = \frac{C(Q)}{\tau} = \frac{C(Q)}{Q / \lambda} = \frac{\lambda C(Q)}{Q}
\]

\[
= \frac{\lambda}{Q} \left( K + c^p Q + c^h \left( \frac{Q^2}{2\lambda} \right) \right)
\]

\[
= \frac{\lambda K}{Q} + \frac{\lambda c^p}{Q} + \frac{c^h Q}{2}
\]

Differentiating with respect to \( Q \) and setting to 0:

\[
\frac{dC^\tau (Q)}{dQ} = -\frac{\lambda K}{Q^2} + \frac{c^h}{2} = 0
\]
The economic order quantity

Finally, solving for $Q$ gives us:

$$Q = \sqrt{\frac{2K\lambda}{c^h}} = \text{The Economic Order Quantity (EOQ)}$$

Also called the Economic Lot Size.

Properties of the optimal solution:

1. Purchase costs do not enter the equation (why?)
2. Order costs per unit time = holding costs per unit time:

   \[
   \text{Order costs per unit time} = \frac{\lambda K}{Q} = \frac{\lambda K}{\sqrt{\frac{2K\lambda}{c^h}}} = \sqrt{\frac{c^h K \lambda}{2}}
   \]

   \[
   \text{Holding costs per unit time} = \frac{c^h Q}{2} = \frac{c^h}{2} \sqrt{\frac{2K\lambda}{c^h}} = \sqrt{\frac{c^h K \lambda}{2}}
   \]
The economic order quantity

- The average cost function:

\[
\text{Total cost} = \sqrt{\frac{c^h K \lambda}{2}}
\]

\[
\text{Order costs} = \frac{2K \lambda}{\sqrt{c^h}}
\]

\[
\text{Holding cost} = \sum \sqrt{\frac{c^h K \lambda}{2}}
\]
The economic order quantity

- Sensitivity analysis
Outline

- Basic inventory problems
- The economic order quantity
- An inventory game
- Multiperiod lot sizing
  » Math programming formulation
  » Heuristics
  » Wagner-Whitin algorithm
An inventory game

- The basics:
  - Random demand (uniform between 0 and 10)
  - Demands are not revealed until an order is entered.
  - Cost parameters:

<table>
<thead>
<tr>
<th>Order cost</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purchase</td>
<td>10</td>
</tr>
<tr>
<td>Holding</td>
<td>1</td>
</tr>
<tr>
<td>Stockout</td>
<td>5</td>
</tr>
</tbody>
</table>
# An inventory game

**Player: Joe**

<table>
<thead>
<tr>
<th>Time</th>
<th>Inventory</th>
<th>Order</th>
<th>Demand</th>
<th>Ending</th>
<th>Sold</th>
<th>Lost</th>
<th>Order</th>
<th>Purchase</th>
<th>Holding</th>
<th>Stock</th>
<th>Total</th>
<th>Cum</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>20</td>
<td>0</td>
<td>2</td>
<td>18</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>18</td>
<td>0</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>1</td>
<td>18</td>
<td>0</td>
<td>1</td>
<td>17</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17</td>
<td>0</td>
<td>17</td>
<td>35</td>
</tr>
<tr>
<td>2</td>
<td>17</td>
<td>0</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>9</td>
<td>44</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>50</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>54</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>2</td>
<td>6</td>
<td>0</td>
<td>25</td>
<td>40</td>
<td>2</td>
<td>0</td>
<td>67</td>
<td>121</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>25</td>
<td>60</td>
<td>4</td>
<td>0</td>
<td>89</td>
<td>210</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>0</td>
<td>25</td>
<td>40</td>
<td>3</td>
<td>0</td>
<td>68</td>
<td>278</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>5</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>25</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>75</td>
<td>353</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>8</td>
<td>6</td>
<td>2</td>
<td>6</td>
<td>0</td>
<td>25</td>
<td>80</td>
<td>2</td>
<td>0</td>
<td>107</td>
<td>460</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>6</td>
<td>2</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>25</td>
<td>60</td>
<td>6</td>
<td>0</td>
<td>91</td>
<td>551</td>
</tr>
<tr>
<td>11</td>
<td>6</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>25</td>
<td>20</td>
<td>5</td>
<td>0</td>
<td>50</td>
<td>601</td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>0</td>
<td>25</td>
<td>30</td>
<td>3</td>
<td>0</td>
<td>58</td>
<td>659</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>0</td>
<td>25</td>
<td>50</td>
<td>3</td>
<td>0</td>
<td>78</td>
<td>737</td>
</tr>
<tr>
<td>14</td>
<td>3</td>
<td>5</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>25</td>
<td>50</td>
<td>8</td>
<td>0</td>
<td>83</td>
<td>820</td>
</tr>
<tr>
<td>15</td>
<td>8</td>
<td>0</td>
<td>2</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>826</td>
</tr>
<tr>
<td>16</td>
<td>6</td>
<td>2</td>
<td>9</td>
<td>0</td>
<td>8</td>
<td>1</td>
<td>25</td>
<td>20</td>
<td>0</td>
<td>5</td>
<td>50</td>
<td>876</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>8</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>25</td>
<td>80</td>
<td>7</td>
<td>0</td>
<td>112</td>
<td>988</td>
</tr>
<tr>
<td>18</td>
<td>7</td>
<td>1</td>
<td>6</td>
<td>2</td>
<td>6</td>
<td>0</td>
<td>25</td>
<td>10</td>
<td>2</td>
<td>0</td>
<td>37</td>
<td>1025</td>
</tr>
<tr>
<td>19</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>25</td>
<td>50</td>
<td>3</td>
<td>0</td>
<td>78</td>
<td>1103</td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>4</td>
<td>8</td>
<td>0</td>
<td>7</td>
<td>1</td>
<td>25</td>
<td>40</td>
<td>0</td>
<td>5</td>
<td>70</td>
<td>1173</td>
</tr>
</tbody>
</table>

|         |           |       |       |       |       |       |       |          |         |       |
| 375     | 680       | 108   | 10    | 1173  |
## An inventory game

### Player Jimmie:

<table>
<thead>
<tr>
<th>Time</th>
<th>Inventory</th>
<th>Order</th>
<th>Demand</th>
<th>Ending</th>
<th>Sold</th>
<th>Lost</th>
<th>Order</th>
<th>Purchase</th>
<th>Holding</th>
<th>Stock</th>
<th>Total</th>
<th>Cum</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>20</td>
<td>0</td>
<td>2</td>
<td>18</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>18</td>
<td>0</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>1</td>
<td>18</td>
<td>0</td>
<td>1</td>
<td>17</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17</td>
<td>0</td>
<td>17</td>
<td>35</td>
</tr>
<tr>
<td>2</td>
<td>17</td>
<td>0</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>9</td>
<td>44</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>50</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>54</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>20</td>
<td>6</td>
<td>18</td>
<td>6</td>
<td>0</td>
<td>25</td>
<td>200</td>
<td>18</td>
<td>0</td>
<td>243</td>
<td>297</td>
</tr>
<tr>
<td>6</td>
<td>18</td>
<td>0</td>
<td>4</td>
<td>14</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>14</td>
<td>0</td>
<td>14</td>
<td>311</td>
</tr>
<tr>
<td>7</td>
<td>14</td>
<td>0</td>
<td>5</td>
<td>9</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>9</td>
<td>320</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>0</td>
<td>8</td>
<td>1</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>321</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>20</td>
<td>6</td>
<td>15</td>
<td>6</td>
<td>0</td>
<td>25</td>
<td>200</td>
<td>15</td>
<td>0</td>
<td>240</td>
<td>561</td>
</tr>
<tr>
<td>10</td>
<td>15</td>
<td>0</td>
<td>2</td>
<td>13</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>13</td>
<td>0</td>
<td>13</td>
<td>574</td>
</tr>
<tr>
<td>11</td>
<td>13</td>
<td>0</td>
<td>3</td>
<td>10</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>584</td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>589</td>
</tr>
<tr>
<td>13</td>
<td>5</td>
<td>20</td>
<td>5</td>
<td>20</td>
<td>5</td>
<td>0</td>
<td>25</td>
<td>200</td>
<td>20</td>
<td>0</td>
<td>245</td>
<td>834</td>
</tr>
<tr>
<td>14</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td>854</td>
</tr>
<tr>
<td>15</td>
<td>20</td>
<td>0</td>
<td>2</td>
<td>18</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>18</td>
<td>0</td>
<td>18</td>
<td>872</td>
</tr>
<tr>
<td>16</td>
<td>18</td>
<td>0</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>9</td>
<td>881</td>
</tr>
<tr>
<td>17</td>
<td>9</td>
<td>0</td>
<td>1</td>
<td>8</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>889</td>
</tr>
<tr>
<td>18</td>
<td>8</td>
<td>0</td>
<td>6</td>
<td>2</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>891</td>
</tr>
<tr>
<td>19</td>
<td>2</td>
<td>10</td>
<td>4</td>
<td>8</td>
<td>4</td>
<td>0</td>
<td>25</td>
<td>100</td>
<td>8</td>
<td>0</td>
<td>133</td>
<td>1024</td>
</tr>
<tr>
<td>20</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1024</td>
</tr>
</tbody>
</table>

100  700  224  0  1024
## An inventory game

### Player:

<table>
<thead>
<tr>
<th>Time</th>
<th>Inventory</th>
<th>Order</th>
<th>Demand</th>
<th>Ending</th>
<th>Sold</th>
<th>Lost</th>
<th>Order</th>
<th>Purchase</th>
<th>Holding</th>
<th>Stocked</th>
<th>Total</th>
<th>Cum</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>

© 2013 W.B. Powell
## An inventory game

<table>
<thead>
<tr>
<th>Time</th>
<th>Quantity</th>
<th>Order</th>
<th>Demand</th>
<th>Ending</th>
<th>Sold</th>
<th>Lost</th>
<th>Order</th>
<th>Purchase</th>
<th>Holding</th>
<th>Stock</th>
<th>Total</th>
<th>Cum</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>6</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>9</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>11</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>12</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>13</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>14</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>15</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>16</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>17</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>18</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>

© 2013 W.B. Powell
Outline

- Basic inventory problems
- The economic order quantity
- An inventory game
- Multiperiod lot sizing
  - Math programming formulation
  - Heuristics
  - Wagner-Whitin algorithm
Math programming formulation

- What do we do when the demands are nonstationary?
  - \( D_t = \) the forecasted demand for time period \( t, 0 \leq t < T \).
  - We are going to use a point estimate of the demand, which produces a deterministic model.
  - \( T = \) the planning horizon.

- Objective function:

\[
\min_{x_t, y_t} \sum_{t=0}^{T-1} c_t \left( R_t, x_t, y_t \right)
\]
Math programming formulation

- The optimization problem can be visualized as a network:

$$\sum_{t=0}^{T-1} D_t$$

Production arcs

Consumption

Inventory

© 2013 W.B. Powell
Math programming formulation

- Integer programming formulation:

  Notation:
  Activity variables:
  \[ R_t = \text{inventory at beginning of period } t \]
  \[ D_t = \text{demand during period starting at } t \]

  Parameters:
  \[ c^h = \text{Unit holding cost per time period} \]
  \[ c^p = \text{Unit purchase cost} \]
  \[ K = \text{Fixed order cost} \]

  Decision variables:
  \[ x_t = \text{amount ordered in period } t \]
  \[ y_t = \begin{cases} 
1 & x_t > 0 \\
0 & x_t = 0 
\end{cases} \]
Math programming formulation

Objective function:

$$\min_{x,y} \sum_{t=0}^{T-1} K y_t + c^p_t x_t + c^h \left( R_t + x_t - D_t \right)$$

subject to:

$$R_{t+1} = R_t + x_t - D_t$$

$$x_t \leq My_t \quad (M = \text{big number})$$

$$x_t \geq D_t - R_t$$

$$x_t \geq 0$$

$$y_t = (0,1)$$

This is an integer programming problem, which can be solved using commercial solvers such as Cplex and Gurobi.
Math programming formulation

- We implement our math programming formulation as a *rolling horizon procedure*
  - Optimize over 0-4, implement time 0
  - Roll to time 1, see new information, solve updated problem for time periods 1-5:
  - Roll to time 2, see new information, solve updated problem for time periods 2-6:
Math programming formulation

Rolling horizon procedures

> These are deterministic approximations of the problem over a planning horizon $H$

Objective function:

$$\min \sum_{t'=t}^{t+H} Ky_{tt'} + c^p_{tt'} x_{tt'} + c^h (R_{tt'} + x_{tt'} - D_{tt'})$$

where $x_t = (x_{tt'})_{t'=t,...,t+H}$, $y_t = (y_{tt'})_{t'=t,...,t+H}$

subject to:

$$R_{t,t'+1} = R_{tt'} + x_{tt'} - D_{tt'}$$

$$x_{tt'} \leq My_{tt'} \quad (M = \text{big number})$$

$$x_{tt'} \geq D_{tt'} - R_{tt'}$$

$$x_{tt'} \geq 0$$

$$y_{tt'} = (0,1)$$
Heuristics

Silver-Meal heuristic (Least average cost)

Let $C(s) =$ average cost per unit time if we order over the next $s$ time periods

$$
= \frac{1}{s} \left( K + c^h \sum_{t=0}^{s-1} tD_t \right)
$$

Calculate $C(1), C(2), \ldots, C(s)$. Stop when $C(s + 1) > C(s)$. Set $T = s$. Order enough for the next $T$ time periods.

» One of the best known and most widely used heuristics in supply chain management.
Heuristics

■ Least unit cost

Let $C(s) =$ average cost per unit produced if we order over the next $s$ time periods

$$C(s) = \frac{K + c^h \sum_{t=0}^{s-1} tD_t}{\sum_{t=0}^{s-1} D_t}$$

Calculate $C(1)$, $C(2)$, ..., $C(s)$. Stop when $C(s + 1) > C(s)$.
Set $T = s$.

Least unit cost reflects the way managers are actually measured. No one is measured in terms of $$/$/day (why not??).
Wagner-Whitin algorithm

Example: Seasonal TV demand

» Parameters:

<table>
<thead>
<tr>
<th>Setup cost</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order cost</td>
<td>1</td>
</tr>
<tr>
<td>Holding cost</td>
<td>1.3</td>
</tr>
</tbody>
</table>

» Demands (in 1000’s):

<table>
<thead>
<tr>
<th>Time</th>
<th>Demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>
Wagner-Whitin algorithm

A network representation:

» The set of decisions represents a shortest path problem over a specialized network:

» The cost on each arc is the cost of the decision, including order costs and all holding costs.

» What does the optimal solution look like?
Wagner-Whitin algorithm

Properties of optimal solution

» We only need to make decisions when the inventory is zero.

» This means our decision variable is not the quantity, but the number of time periods into the future that we need to cover.

Order enough just for time periods 0 and 1

Order enough just for time periods 2 and 3

The state that we are at time period 2 with zero inventory.
Wagner-Whitin algorithm

Start with the final node:

\[ \begin{align*}
43.7 & \quad 29.7 & \quad 17.6 & \quad 10 & \quad 0 \\
\end{align*} \]
Wagner-Whitin algorithm

Links into time period 3:
Wagner-Whitin algorithm

- Links into time period 2:
Wagner-Whitin algorithm

Links into time period 1:
Wagner-Whitin algorithm

Finally, we walk forward in time:

We use the values computed in the backward pass to walk forward and compute decisions.
Wagner-Whitin algorithm

Strengths:

» Very fast

» Handles very general cost functions
  • You can use virtually any shape order cost function.

» Handles time-dependent data (e.g. seasonal data, day of week effects or hour of day patterns).

» Handles forecasts of the future (which is a form of time-dependency).
Wagner-Whitin algorithm

Limitations of this model:
» Assumes demands are deterministic!!!
  - “Optimal” solution is not really optimal.
  - Drives inventories to zero, which will create stockouts.
  - Have to reoptimize as forecasted demands change.

» Limitations:
  - Computationally demanding when you have to solve 100,000 problems (Wal-Mart!).
  - Solutions are not “obviously” better than good heuristics under realistic conditions.
  - Gets complicated if you have multiple items and joint capacity constraints (need to use integer programming formulation)

» But:
  - Serves as a useful subproblem in the context of larger applications.
  - Highlights behavior of the problem.
Wagner-Whitin algorithm

- Important generalizations:
  - Upper bounds on order quantities
    - What if we cannot order more than \( u_t \) in time period \( t \)?
  - Upper bounds on production and multiple items:
    - This is the problem that actually arises in practice.
    - Called the “capacitated multi-item lot sizing problem.”
  - Limit on total production time, in the presence of setup times.
    - The literature on setup times is very sparse.