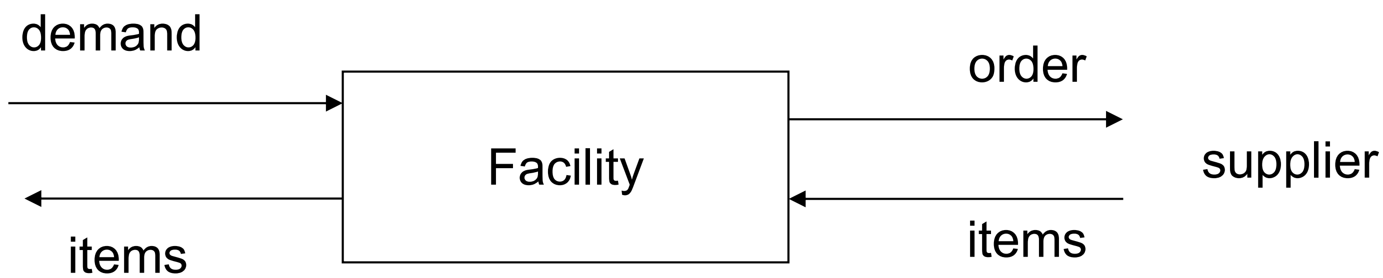
Simulation Model 2

A Simple Inventory System

Conceptual Model



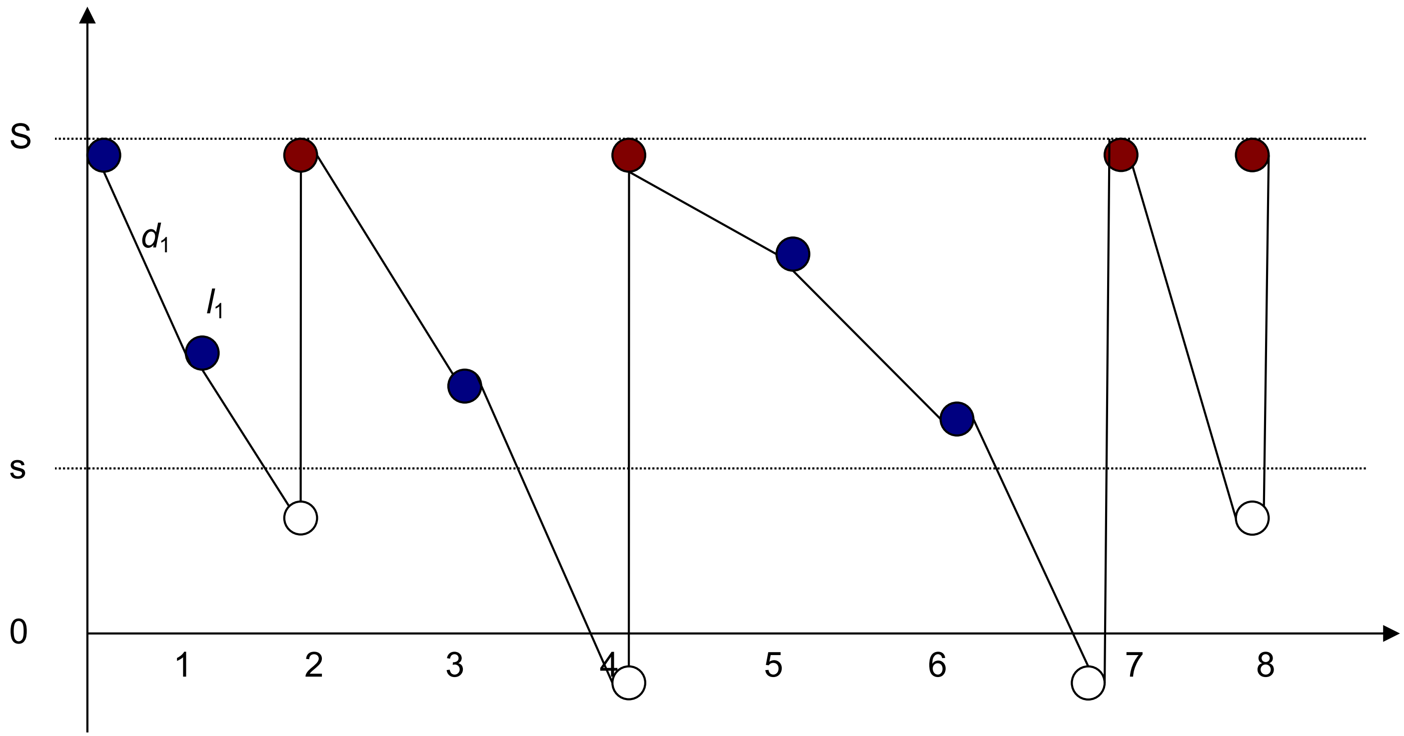
<u>Policies:</u>

- Inventory review is periodic
- Items are ordered, if necessary, only at review times
- Maximum level: S
- Minimum level: s
- Back ordering is possible
- No delivery lag
- Both Initial and terminal inventory level are S

Specification Model

- Time begins at $t = 0$
- Review times are $t = 0, 1, 2, \ldots$
- l_{i-1} : inventory level at beginning of *i* th interval
- o_{i-1} : amount ordered at time $t = i - 1$, ($o_{i-1} >= 0$)
- d_i : demand quantity during *i* th interval, ($d_i >= 0$)
- Inventory at end of interval can be negative

Inventory levels



The Sequences
l_i, o_i, d_i, $\{i=0, 1, 2, \ldots\}$

- At $t = i - 1$: l_{i-1}, o_{i-1}
- $l_{i-1}>=s$ → no order is placed
- $l_{i-1}<s$ → replenished to S
- At the end of the *ith* interval → Items are delivered immediately, inventory diminished by d_i.

$$o_{i-1} = \begin{cases} 0, & \text{if } l_{i-1} \geq s \\ S\text{-}\, l_{i\text{-}1}, & \text{if } l_{i-1} < s \end{cases}$$

$$l_i = l_{i-1} + o_{i-1} - d_i$$

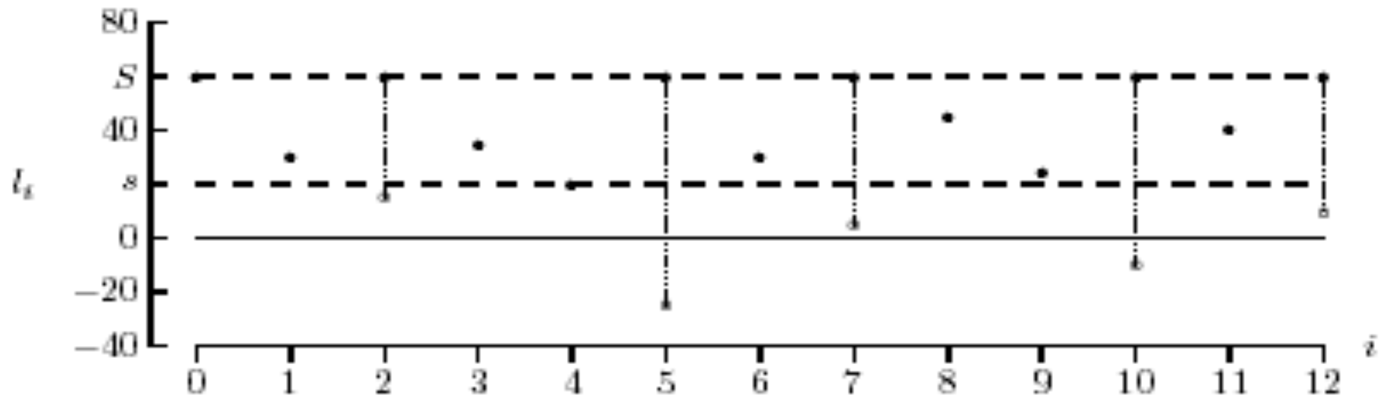
Time Evolution of Inventory Level

$l_0 = S$;   /* the initial inventory level is S */
$i = 0$;
while (more demand to process)
{
    i++;
    if ($l_{i-1} < s$)
        { $o_{i-1} = S - l_{i-1}$; }
    else
        {$o_{i-1} = 0$;}
    d_i = GetDemand();
    $l_i = l_{i-1} + o_{i-1} - d_i$;
}
$n = i$;
$o_n = S - l_n$;
$l_n = S$;   /* the terminal inventory level is S */
return $l_1, l_2, \ldots, l_n$ and $o_1, o_2, \ldots, o_n$;

Sample Demands

Let (s, S) = (20, 60) and consider n = 12 time intervals:

i :	1	2	3	4	5	6	7	8	9	10	11	12
di :	30	15	25	15	45	30	25	15	20	35	20	30



Output Statistics

- Average demand and average order
- For Example data

$$\bar{d} = \frac{1}{n}\sum_{i=1}^{n} d_i$$

$$\bar{d} = \bar{o} = \frac{305}{12} \approx 25.42$$

(items per time interval)

$$\bar{o} = \frac{1}{n}\sum_{i=1}^{n} o_i$$

Flow Balance

- Over the simulated period, all demand is satisfied.

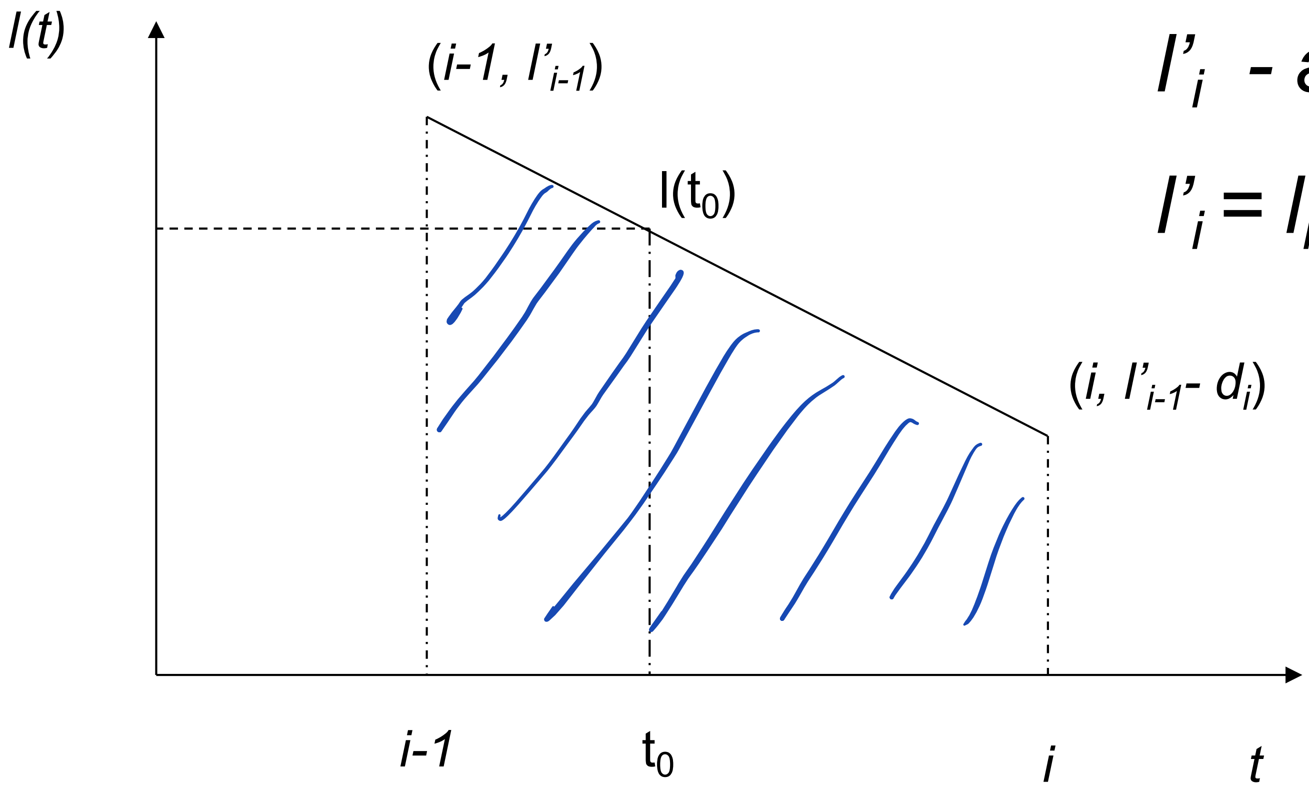
  starting inventory(S) = ending inventory(S)

  average demand ($\bar{d}$) = average items per order ($\bar{o}$)

  average "flow" of items *in* = average "flow" of items *out*

- The inventory system is ***flow balanced***

Inventory Level as a Function of Time - $I(t)$



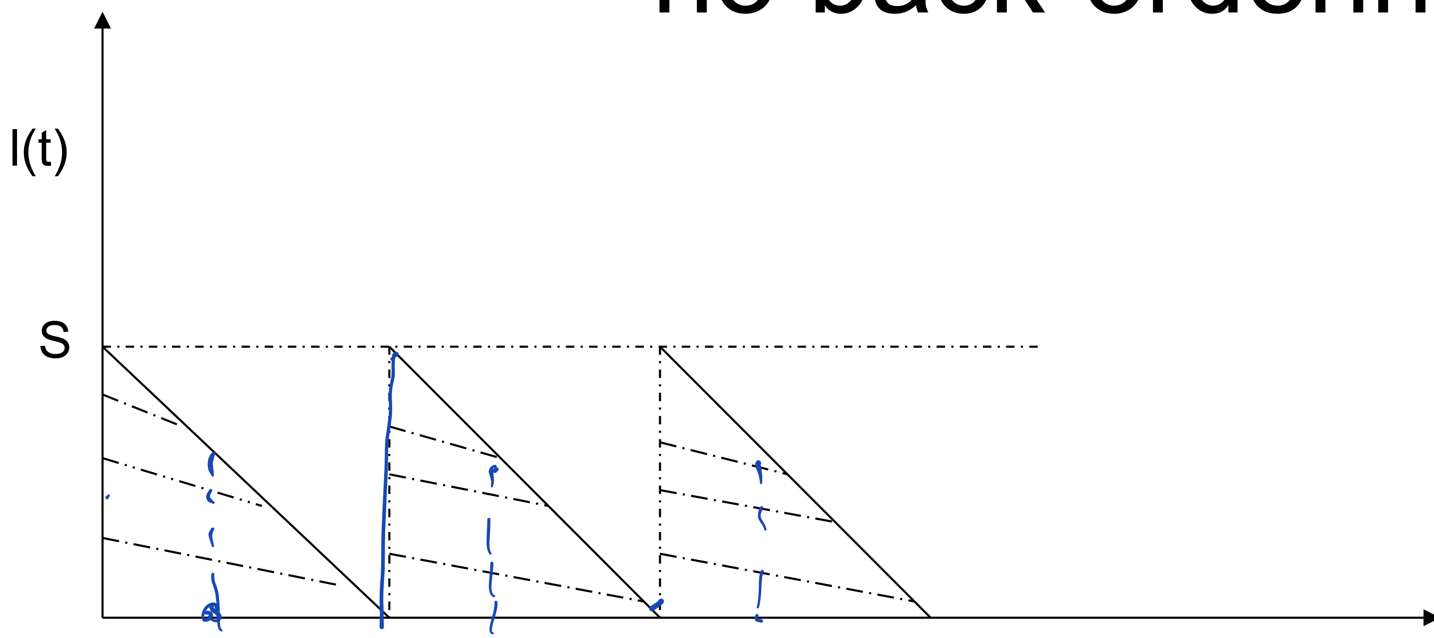
I_i - *before review*

I'_i - *after review*

$I'_i = I_i + o_i$

Constant Demand Rate – Linear function

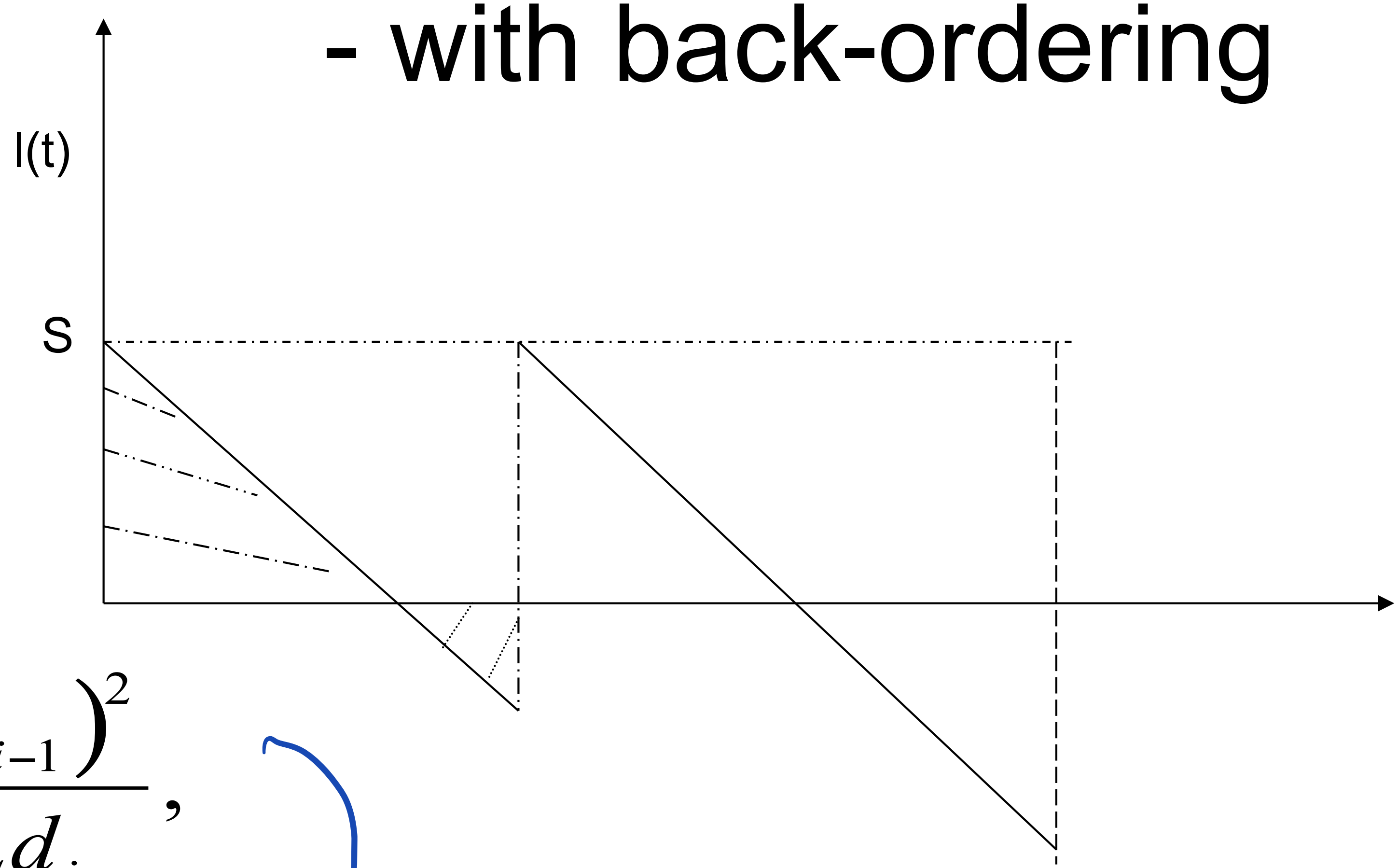
Average Inventory Level
- no back-ordering



$$\bar{l}_i^{\,+} = \frac{l'_{i-1} + (l'_{i-1} - d)}{2} = l'_{i-1} - \frac{1}{2} d_i,$$

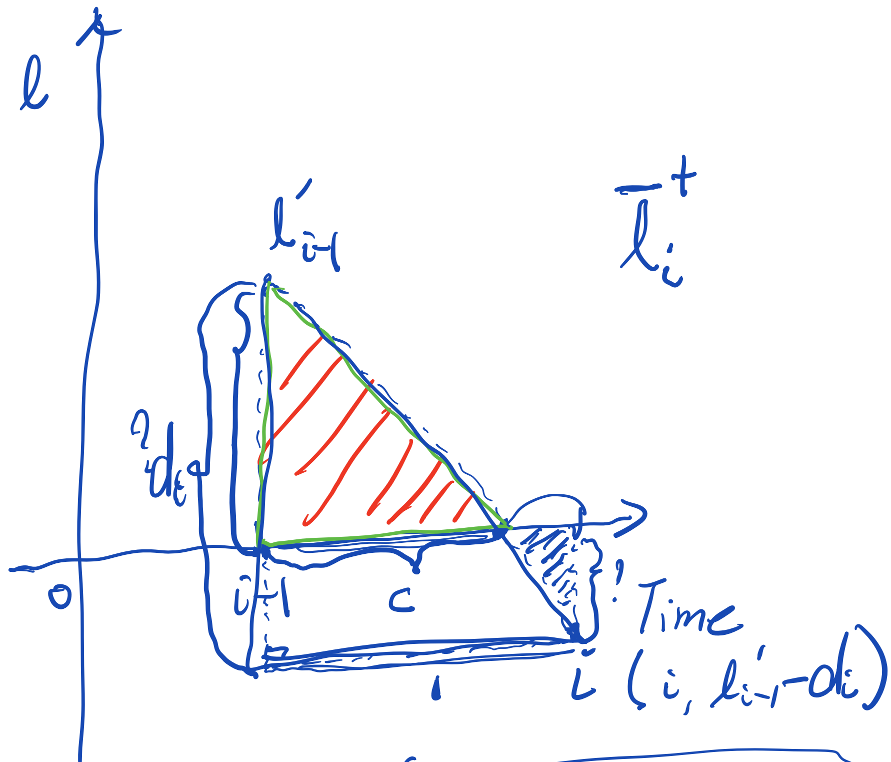
$$\bar{l}_i^{\,-} = 0.$$

Average Inventory Level
- with back-ordering



$$\bar{l}_i^{\,+} == \frac{(l'_{i-1})^2}{2d_i},$$

$$\bar{l}_i^{\,-} = \frac{(d_i - l'_{i-1})^2}{2d_i}.$$



$$\bar{l}_i^+ = ? \qquad \boxed{\frac{c}{1} = \frac{l'_{i-1}}{d_i}}$$

$$c = \frac{l'_{i-1}}{d_i}$$

$$\bar{l}_i^+ = \frac{1}{2} \cdot c \cdot l'_{i-1} = \frac{1}{2} \cdot \frac{l'_{i-1}}{d_i} \cdot l'_{i-1}$$

$$= \frac{1}{2} \frac{l'^{2}_{i-1}}{d_i}$$

$$\bar{l}_i^- = \frac{1}{2} (1-c) \cdot (d_i - l'_{i-1})$$

$$= \frac{1}{2} \left(1 - \frac{l'_{i-1}}{d_i}\right)(d_i - l'_{i-1})$$

$$(d_i - l'_{i})^2$$

$= \frac{(d_i - \ell_{i-1})}{2 d_i}$

Time-Averaged Inventory Level

- Time-averaged holding level and time-averaged shortage level:

$$\bar{l}^{+} = \frac{1}{n}\sum_{i=1}^{n} \bar{l}^{+}_{i}, \qquad \bar{l}^{-} = \frac{1}{n}\sum_{i=1}^{n} \bar{l}^{-}_{i}$$

- Note that time-averaged shortage level is positive
- The time-averaged inventory level is:

$$\bar{l} = \bar{l}^{+} - \bar{l}^{-}$$

Computational Model

- sis1.c
- Computes the statistics

$$\bar{d},\ \bar{o},\ \bar{l}^{+},\ \bar{l}^{-} \quad \text{and} \quad \bar{u} = \frac{\text{number of orders}}{n}$$

$\bar{u}$ is the order frequency.

Operating Costs

- A facility's cost of operation is determined by:

C-item : unit cost of new item
C-setup : fixed cost for placing an order
C-hold : cost to hold one item for one time interval
C-short : cost of being short one item for one time interval

Case Study

- Automobile dealership that uses weekly periodic inventory review
- The facility is the showroom and surrounding areas
- The items are new cars
- The supplier is the car manufacturer
- Customers are people …
- *Simple (one type of car) inventory system*

Case Study

- Limited to a maximum of S = 80 cars
- Inventory reviewed every Monday
- If inventory falls below s = 20, order cars sufficient to restore to S
- Ignore delivery lag
- Costs:
- Item cost is C-item = $8000 per item
- Setup cost is C-setup = $1000
- Holding cost is C-hold = $25 per week
- Shortage cost is C-short = $700 per week

Per-Interval Average Operating Costs

- The average operating costs per time interval are
  - item cost : C-item * (average) o $\bar{o}$
  - setup cost : C-setup * (average) u
  - holding cost : C-hold * (average) I^+
  - shortage cost : C-short * (average) I^-
- The average total operating cost per time interval is their sum
- For the stats and costs of the hypothetical dealership:
  - item cost : $\$8000 \cdot 29.29 = \$234,320$ per week
  - setup cost : $\$1000 \cdot 0.39 = \$390$ per week
  - holding cost : $\$25 \cdot 42.40 = \$1,060$ per week
  - shortage cost : $\$700 \cdot 0.25 = \$175$ per week

Cost Minimization

- By varying s (and possibly S), an optimal policy can be determined.

- Optimal $\Rightarrow$ minimum average cost

  $\overline{o} = \overline{d}$ and $\overline{d}$ depends only on the demands

  $\Rightarrow$ item cost is independent of (s, S)
- Average dependent cost is:

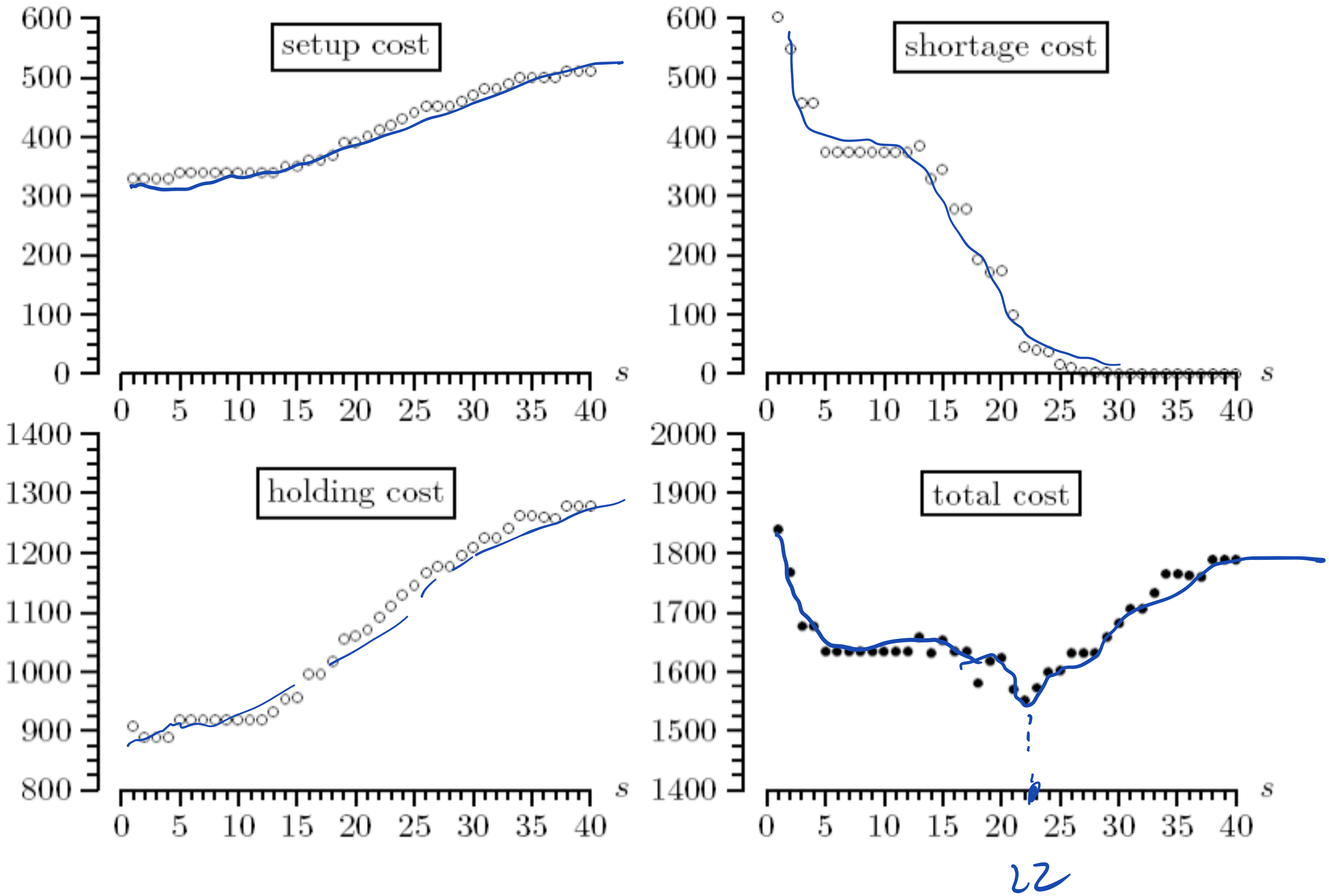
  avg setup cost + avg holding cost + avg shortage cost

Experimentation

- Let S be fixed, and let the demand sequence be fixed.
- If s is systematically increased, we expect:

  - average setup cost and holding cost will increase as *s* increases.
  - average shortage cost will decrease as *s* increases.
  - average dependent cost will have ‘U’ shape, yielding an optimum.

Simulation Results –
minimum cost is $1550 at s = 22



Monte Carlo Simulation

Applications

Definition

- Recall:

System model

Deterministic

Stochastic

Static

$\boxed{\text{Monte Carlo Simulation}}$

Relative Frequency and Probability

- **Relative Frequency**: Perform an experiment many times (n) and count the number of occurrences (n_a) of an event A.

Relative frequency of occurrence of event A

$\rightarrow$ n_a/n.

- Let Pr(A) be the probability of event A. Then

$$\lim_{n \to \infty} \frac{n_a}{n} = \Pr(A)$$

Example 1

Roll two dice and observe the up faces

(1, 1) (1, 2) (1, 3) (1, 4) (1, 5) (1, 6)

(2, 1) (2, 2) (2, 3) (2, 4) (2, 5) (2, 6)

(3, 1) (3, 2) (3, 3) (3, 4) (3, 5) (3, 6)

(4, 1) (4, 2) (4, 3) (4, 4) (4, 5) (4, 6)

(5, 1) (5, 2) (5, 3) (5, 4) (5, 5) (5, 6)

(6, 1) (6, 2) (6, 3) (6, 4) (6, 5) (6, 6)

Estimate Probability

- If the two up faces are summed, an integer-valued random variable, say X, is defined with possible values 2 through 12 inclusive.

sum, x :	2	3	4	5	6	7	8	9	10	11	12
Pr(X = x) :	1/36,	2/36,	3/36,	4/36,	5/36,	6/36,	5/36,	4/36,	3/36,	2/36,	1/36

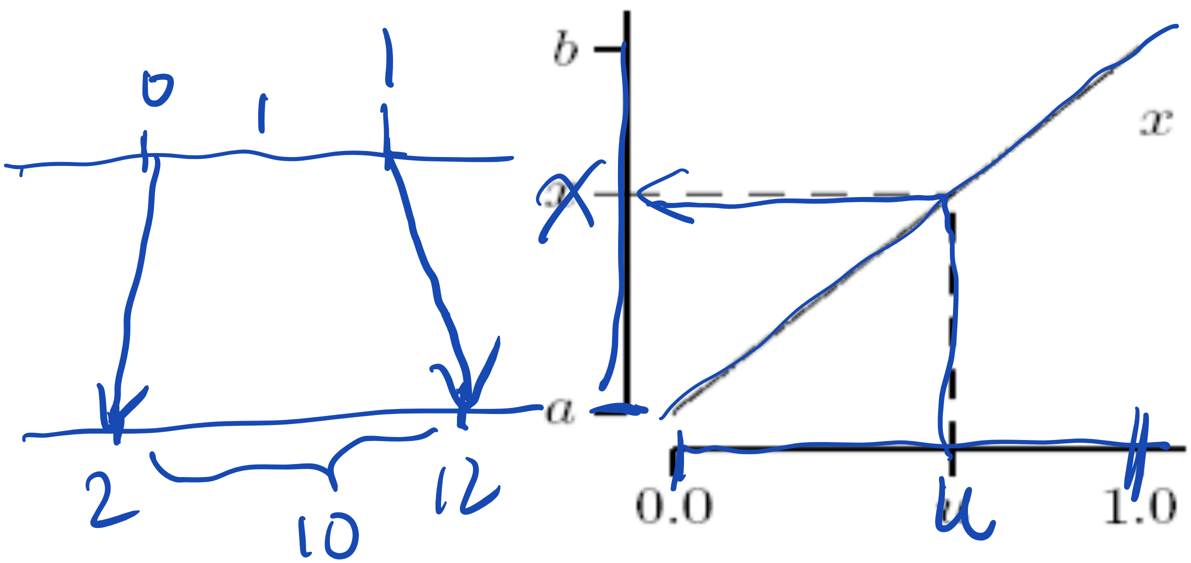
- Pr(X = 7) could be estimated by replicating the experiment many times and calculating the relative frequency of occurrence of 7's

Random Variates

- A Random Variate is an algorithmically generated realization of a random variable
- u = Random() generates a Uniform(0, 1) random variate

$0 < u < 1$

- How to generate Uniform(a, b) variate?



$x = a + (b - a)u$

(a, b)

$(2, 12)$

$x = a + (b-a)u$

$x = 2 + (12-2)u$

Generating a Uniform Random Variate

```
double Uniform(double a, double b)
                                    /* use a < b */
{
        return (a + (b - a) * Random());
}
```

$u \in (0, 1)$

Integer-valued Random Variates

$$x = a + floor\ ((b-a+1)u)$$

$\lfloor (b-a+1)u \rfloor$

```
long Equilikely(long a, long b)
                                  /* use a < b */
{
  return (a + (long) ((b - a + 1) * Random()));
}
```

$a = 1, \quad b = 6$

$1, 2, 3, 4, 5, 6$

Examples

- To generate a random variate *x* that simulates rolling two fair dice and summing the resulting up faces, use

  x = Equilikely(1, 6) + Equilikely(1, 6);
- Note that this is not equivalent to

  x = Equilikely(2, 12);

- To select an element x at random from the array

a[0], a[1], . . ., a[n − 1]:

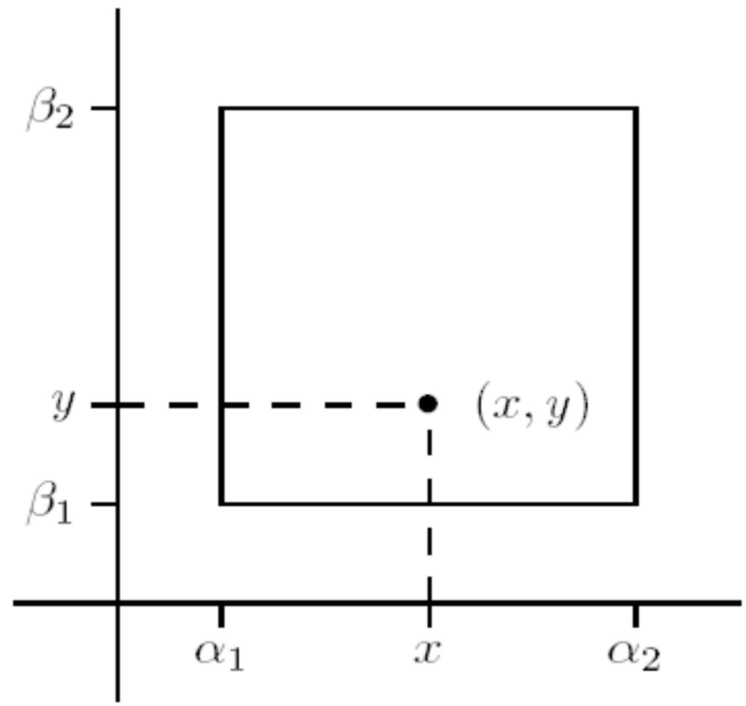
```
i = Equilikely(0, n - 1);  // pick an array index at random
x = a[i];
```

Geometric Applications

- Generate a point at random inside a rectangle with opposite corners at (α_1, β_1) and (α_2, β_2):

x = Uniform(α_1, α_2);

y = Uniform(β_1, β_2);



The value of π

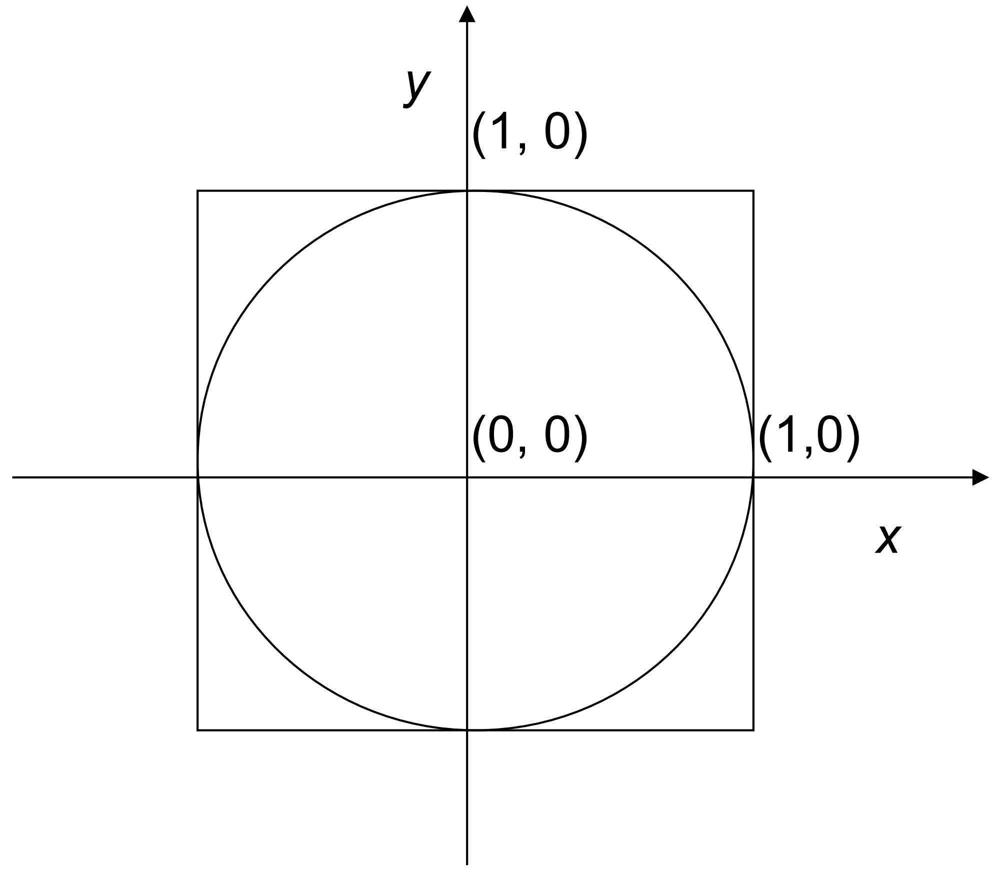
- Area of the unit circle: $A=\pi$
- Area of the unit square = 4
- $P(x, y) \rightarrow$ a random point,

  $-1<x<1, \ -1<y<1,$

The probability for P to fall inside the circle Pr can be estimated as:

$$Pr = \pi/4$$

The value of π can be estimated by simulation program.



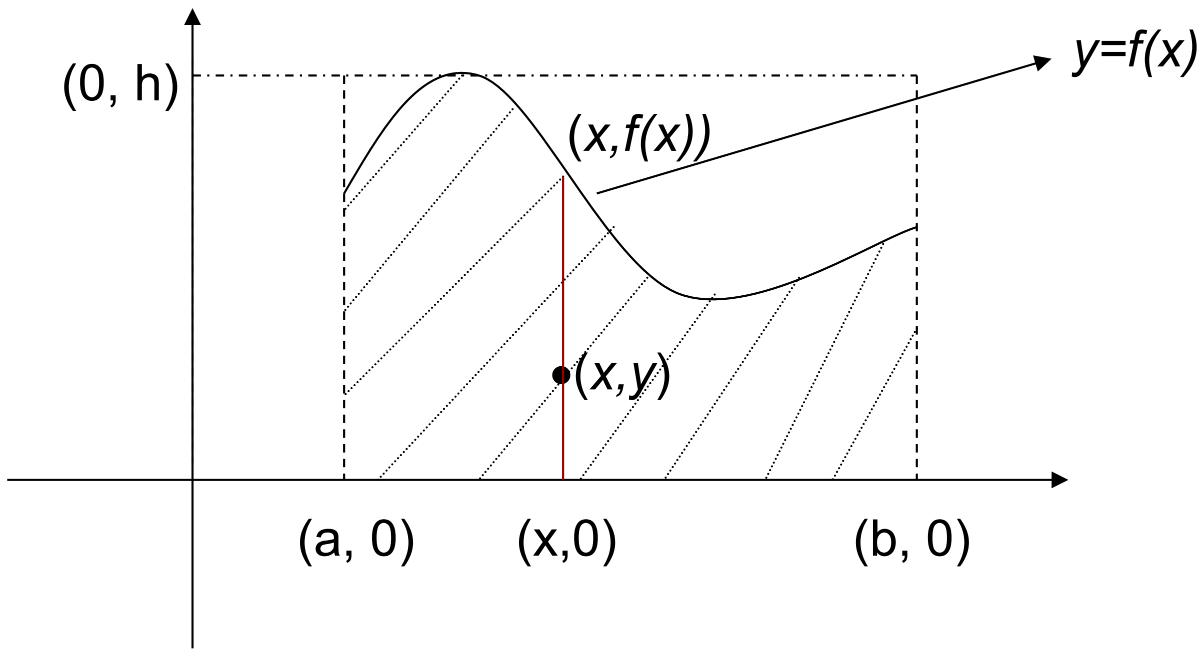
Integration - Estimation

- P(x, y) $\rightarrow$ a random point,

  a<x<b, 0<y<h,

- The probability for P to fall on the shaded region:

  y<f(x)



$$\Pr = \frac{\text{Area under the curve}}{\text{Area of the rectangle}} = \frac{\int_a^b f(x)dx}{(b-a)h}$$