

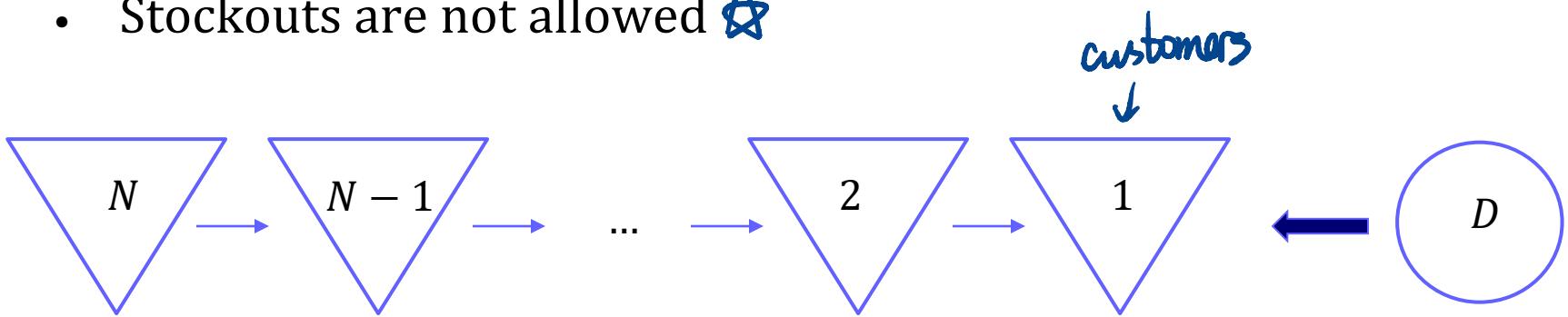
# **LEC009 Inventory Management IV**

**VG441 SS2021**

Cong Shi  
Industrial & Operations Engineering  
University of Michigan

# Multi-Echelon Problem

- $N$  stages of a (serial) supply chain
- Demand rate is  $\lambda$  deterministically only at stage 1
- Stockouts are not allowed  $\star$



$k_j$  = fixed cost for orders of stage  $j$

$h'_j$  = inventory holding-cost rate for stage  $j$

$I'_j(t)$  = (local) inventory of stage  $j$  at time  $t$

$L'_j$  = leadtime for stage- $j$  orders = 0 W.L.O.G.

*the amount of time for goods to lead to*

Question: what is the optimal ordering strategy?

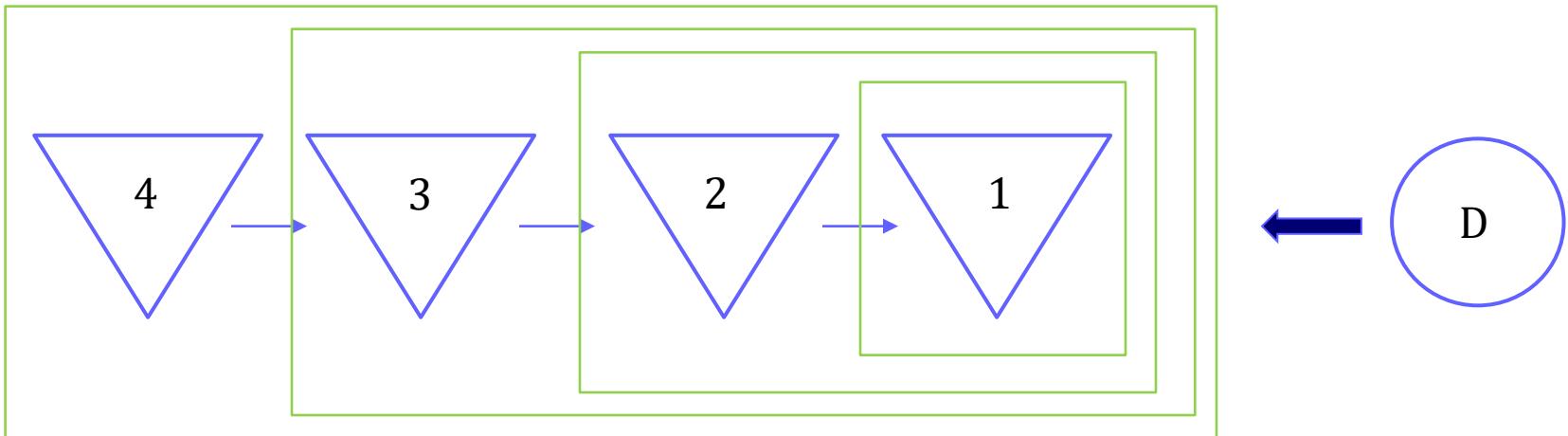
# Echelons and Echelon Inventories

等級

$k_j$  = fixed cost for orders of stage  $j$

$h'_j$  = inventory holding-cost rate for stage  $j$

$I'_j(t)$  = (local) inventory of stage  $j$  at time  $t$



$$I_j(t) = \text{echelon inventory of stage } j \text{ at time } t = \sum_{i \leq j} I'_i(t)$$

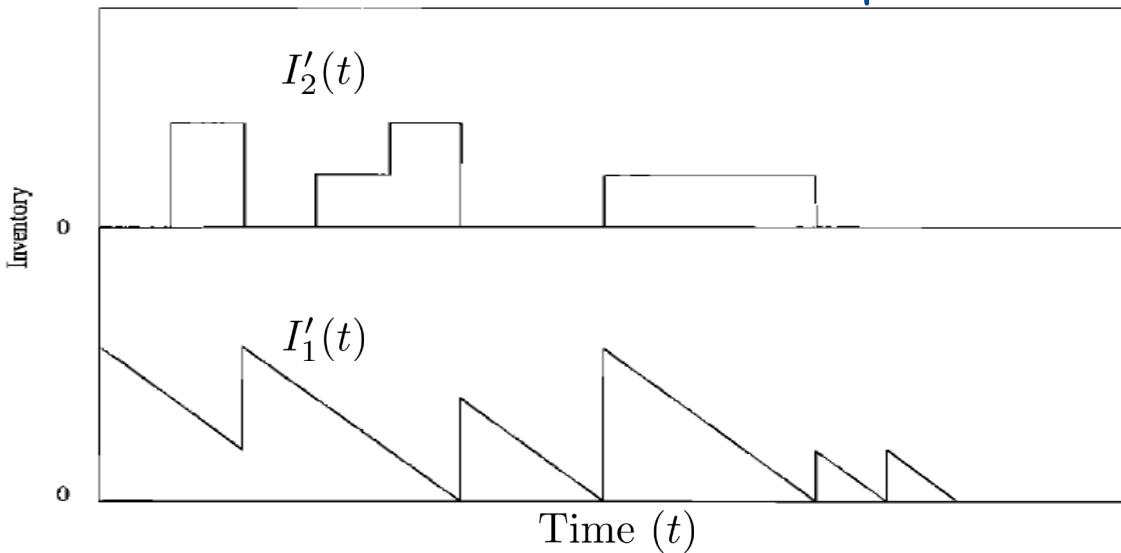
$$h_j = \text{echelon-inventory holding-cost rate for stage } j = h'_j - h'_{j+1} \geq 0$$

$$\sum_j h'_j I'_j(t) = \sum_j h_j I_j(t) \quad \text{for all } t$$

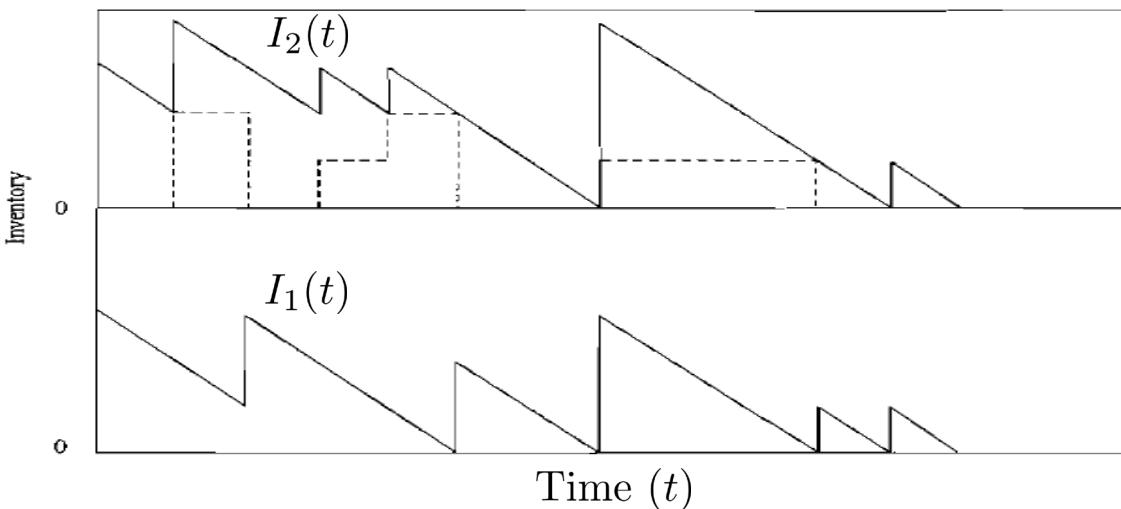
*much more expensive  
to hold downstream than  
upstream*

# Why Echelon Inventories?

Local inventories over time 上游通常不太规律



Echelon inventories over time



# Policy Structures

- A policy is **nested** if for all  $j$ , whenever stage  $j$  orders, so does stages  $j - 1$ . *when we order something in j, then we have to order all things in j-1*
- A policy is **ZIO** (zero-inventory-ordering) if order occurs only when its echelon inventory is zero.

Convince yourself that nested and ZIO are optimal.

# Stationary-Interval Policies

$u_j$  = order interval for stage  $j$

$\mathbf{u}$  = the vector  $(u_j)_j$  ( $u_1, u_2, \dots, u_r$ )

$g_j = h_j \lambda$  ~~demand rate~~

$C(\mathbf{u})$  = average cost of the policy specified by  $\mathbf{u}$



fixed cost portion

$$C(\mathbf{u}) = \sum_j \left[ \frac{k_j}{u_j} + \frac{1}{2} g_j u_j \right]$$

holding cost portion

Minimize  $C(\mathbf{u})$

subject to  $u_j = \xi_j u_{j-1}$ , for all  $j = 2, \dots, J$

$\xi_j \in \mathbb{Z}^+$ , for all  $j = 2, \dots, J$

Caveat: This MILP may be difficult to solve.

NP-hard

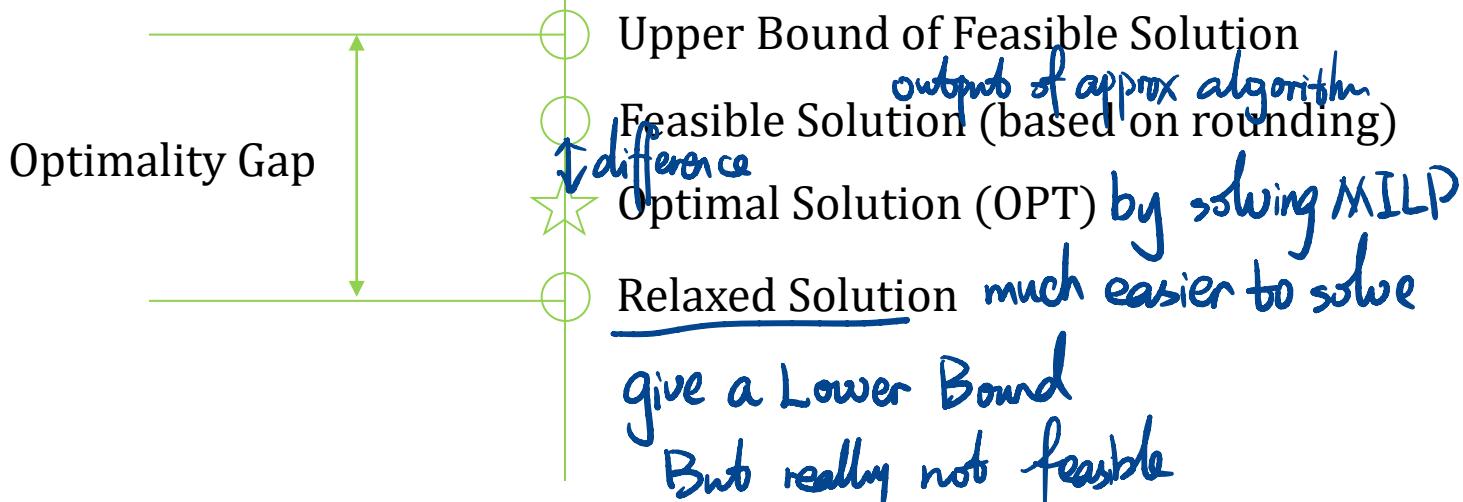
# Here is the Plan

general plan for any  
"go-to-template" approx algorithm

- First, we solve a simpler “relaxed” problem.
- Solution of the relaxed problem is a lower bound.
- We round off this relaxed solution to obtain a feasible solution.
- We get an upper bound on this feasible solution.
- Show that the two bounds are close together.

逐漸逼近法

difference tells you the difference of solution



# Stationary-Interval Policies

$u_j$  = order interval for item  $j$

$\mathbf{u}$  = the vector  $(u_j)_j$

$g_j = h_j \lambda$

$$C(\mathbf{u}) = \sum_j \left[ \frac{k_j}{u_j} + \frac{1}{2} g_j u_j \right]$$

Minimize  $C(\mathbf{u})$

subject to  $u_j = \xi_j u_{j-1}, \quad \text{for all } j = 2, \dots, J$

$\xi_j \in \mathbb{Z}^+, \quad \text{for all } j = 2, \dots, J$

Relax the constraints ...

$\xi_j$  could be any fractional value

Minimize  $C(\mathbf{u})$

subject to  $u_j \geq u_{j-1}, \quad \text{for all } j = 2, \dots, J$

why does this gives a lower bound?

因为 relax constraints  $\Rightarrow$  feasible region becomes larger

$\Rightarrow$  smaller objective value

# Relaxed and Feasible Solutions

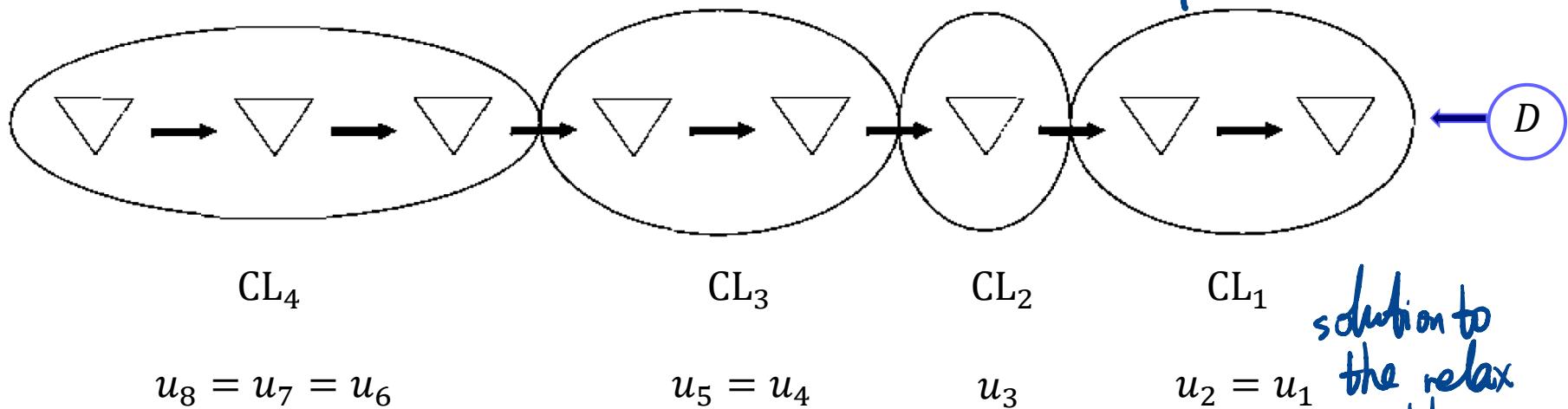
Minimize  
subject to

$C(\mathbf{u})$  project gradient descent

$$u_j \geq u_{j-1}, \quad \text{for all } j = 2, \dots, J$$

power-of-2 cost  
optimal  $\leq 1.06$

Clusters.



Feasible solution: round to the nearest power-of-2

$$\theta_2 = \theta_1 = L \log u_2 \quad (\text{LB})$$

$$\hat{u}_2 = \hat{u}_1 = 2^{\theta_2} = 2^{\theta_1}$$

Gap between this feasible solution and the relaxed solution:  $\sim 1.06$

$\theta \hat{u}_2 = \hat{u}_1$  satisfy integer

# Deterministic Demand (Nonstationary)

Input:  $T$  period demands  $d_1, \dots, d_T$

Decisions:  $q_j(t), I_j(t) \rightarrow$  inventory  
order quantity

binary programme  
use big-M trick.  
 $q_{j,t} \leq M z_{j,t}$   
 $z_j \in \{0,1\}$

$$\begin{array}{ll}\min & \sum_j \sum_t I_j(t) h_j + \sum_j \sum_t k_j \mathbb{1}_{(q_j(t) > 0)} \\ \text{s.t.} & I_j(t) = I_j(t-1) + q_j(t) - d_t \quad \forall t, \forall j \\ & I_j(t-1) + q_j(t) \geq I_{j-1}(t-1) + q_{j-1}(t) \quad \forall t, \forall j \\ & I_j(t) \geq 0, q_j(t) \geq 0 \quad \forall t, \forall j \\ & I_j(0) = 0 \quad \forall j\end{array}$$

# Dynamic Programming *on graphical model*

- Define  $F(i, s, t)$  as the optimal cost of subproblem defined for stages  $i, \dots, 1$  and periods  $[s, t)$
- ZIO:  $I_i(s - 1) = I_i(t - 1) = 0$
- Nested:  $I_j(s - 1) = I_j(t - 1) = 0$ , for all  $j < i$
- Goal:  $F(N, 1, T + 1)$
- Boundary Conditions:

$$F(0, \cdot, \cdot) = 0$$

$$F(\cdot, T + 1, \cdot) = 0$$

$$F(i, s, s + 1) = \sum_{j=1}^i k_j \text{ for all } i$$

*order  
in adjacent for stage j*

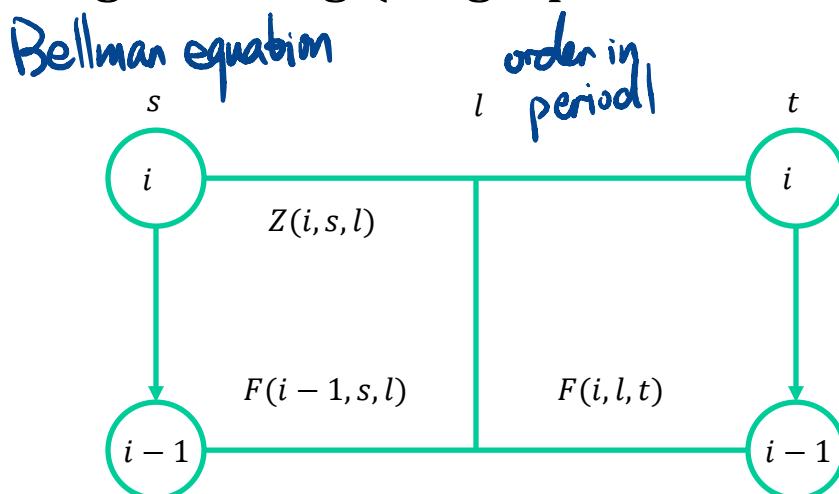
# DP (Graphical Model)

- For each time  $s < t$  and stage  $i$ , the cost of covering  $d_s, \dots, d_{t-1}$

$$Z_{s,t}^i = k_i + \sum_{a=s}^{t-2} h_a \sum_{b=a-1}^{t-1} d_b$$

fixed cost  
holding cost of ordering  $d_{s+1} + d_{s+2} + \dots + d_{t-1}$   
at the begining of  $s$

- Dynamic Programming (on graphical models)



$$F(i, s, t) = \min_{s \leq l \leq t-1} \{ Z_{s,l}^i + F(i-1, s, l) + F(i, l, t) \}$$

immediate    ↓    cost-to-go  
for sub  
stages    cost-to-go  
subsequent  
time

# Stochastic Model 可能性

- Stage 1 faces stochastic demand  $D$  per period and penalty  $p$
- Lead times  $L_1, \dots, L_N$  and lead time demand  $D_1, \dots, D_N$
- Echelon base-stock policy is optimal.

**Theorem:** Let  $\underline{g}_0(x) = (p + h'_1)x^-$ . For  $j = 1, \dots, N$ , let

accelerated

$$\hat{g}_j(x) = h_j x + \underline{g}_{j-1}(x)$$

$$g_j(y) = \mathbb{E}[\hat{g}_j(y - D_j)]$$

$$S_j^* = \operatorname{argmin}\{g_j(y)\}$$

$$\underline{g}_j(x) = g_j(\min\{S_j^*, x\})$$

order up  $E_3^*$

sum inventory ①②③  
 $= E_3^*$

Then  $\mathbf{S}^* = (S_j^*)_{j=1}^N$  is the optimal base-stock vector and  $g_N(S_N^*)$  is the corresponding optimal cost

# Stochastic Model

- Use the following heuristics:

**Theorem 6.4 (Shang and Song (2003)):** For any  $j$  and  $y$

- (a)  $g_j^l(y) \leq g_j(y) \leq g_j^u(y)$
- (b)  $S_j^l \leq S_j^* \leq S_j^u$

$$S_j^l = \tilde{F}_j^{-1} \left( \frac{p + \sum_{i=j+1}^N h_i}{p + \sum_{i=1}^N h_i} \right) \quad \leq \quad S_j^* \quad \leq \quad S_j^u = \tilde{F}_j^{-1} \left( \frac{p + \sum_{i=j+1}^N h_i}{p + \sum_{i=j}^N h_i} \right)$$

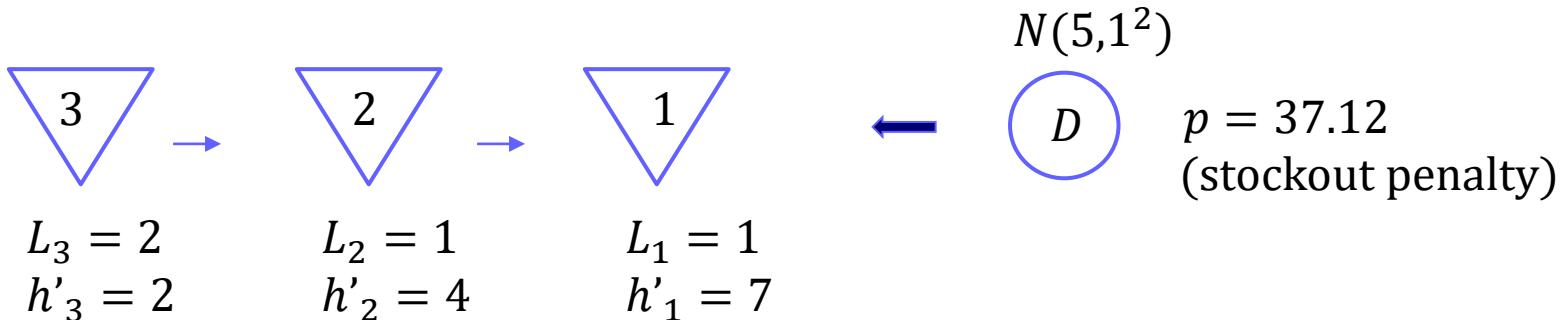
$$\tilde{D}_j = \sum_{i=1}^j D_i$$

↓

Lead-time demand with lead time  $\sum_{i=1}^j L_i$

Computational optimality gap  $\leq 1\%$

# A Simple Example



$$\begin{aligned}\tilde{D}_1 &\sim N(5 \cdot 1, 1^2 \cdot 1) = N(5, 1) \\ \tilde{D}_2 &\sim N(5 \cdot 2, 1^2 \cdot 2) = N(10, 2) \\ \tilde{D}_3 &\sim N(5 \cdot 4, 1^2 \cdot 4) = N(20, 4)\end{aligned}$$

We have  $(h_1, h_2, h_3) = (3, 2, 2)$ . Therefore:

$$\begin{array}{ll} S_1^u = \tilde{F}_1^{-1}\left(\frac{37.12+4}{37.12+7}\right) = 6.49 & S_1^l = \tilde{F}_1^{-1}\left(\frac{37.12+4}{37.12+7}\right) = 6.49 \\ S_2^u = \tilde{F}_2^{-1}\left(\frac{37.12+2}{37.12+4}\right) = 12.35 & S_2^l = \tilde{F}_2^{-1}\left(\frac{37.12+2}{37.12+7}\right) = 11.71 \\ S_3^u = \tilde{F}_3^{-1}\left(\frac{37.12+0}{37.12+2}\right) = 23.27 & S_3^l = \tilde{F}_3^{-1}\left(\frac{37.12+0}{37.12+7}\right) = 22.00 \end{array}$$

Taking the mean, we have

$$\begin{aligned}\tilde{S}_1 &= \frac{1}{2}(6.49 + 6.49) = 6.49 \\ \tilde{S}_2 &= \frac{1}{2}(12.35 + 11.71) = 12.03 \\ \tilde{S}_3 &= \frac{1}{2}(23.27 + 22.00) = 22.63\end{aligned}$$

These values are very close to  $\mathbf{S}^* = (6.49, 12.02, 22.71)$  and indeed their costs are very similar:  $g(\tilde{\mathbf{S}}) = 47.66$ , compared to  $g(\mathbf{S}^*) = 47.65$