MDP Model of Inventory Replenishment

Dept of Industrial Engineering
The University of Iowa

- Consider a periodic-review inventory replenishment system where the maximum allowable inventory level is 7 and up to 3 units may be backordered.
- The daily demand is random, with Poisson distribution having mean of 3 units.
- The inventory on the shelf (the *state*) is counted at the end of each business day, and a *decision* is then made to raise the inventory level to *S* at the beginning of the next business day.
- There is a fixed cost $A=10$ of placing an order, a holding cost $h=1$ for each item in inventory at the end of the day, and a penalty $p=5$ for each unit backordered.

What is desired is a replenishment policy to determine the optimal value of *S* for each possible inventory level.

Probability Distribution of Demand: (Poisson, with mean 3)

d	0	1	2	3	4	5	6	7	8	9	10
P{D=d}	0.0498	0.1494	0.224	0.224	0.168	0.1008	0.0504	0.0216	0.0081	0.0027	0.0008

COSTS as a function of state s and action a: sum of

Storage cost: hs^+

Shortage cost: ps^-

Ordering cost: A if a>s; 0 otherwise

State \ Action	SOH = 0	SOH = 1	SOH = 2	SOH = 3	SOH = 4	SOH = 5	SOH = 6
BO = 3	55	55	55	55	55	55	55
BO = 2	30	30	30	30	30	30	30
BO = 1	15	15	15	15	15	15	15
SOH = 0	0	10	10	10	10	10	10
SOH = 1	∞	1	11	11	11	11	11
SOH = 2	∞	∞	2	12	12	12	12
SOH = 3	∞	∞	∞	3	13	13	13
SOH = 4	∞	∞	∞	∞	4	14	14
SOH = 5	∞	∞	∞	∞	∞	5	15
SOH = 6	∞	∞	∞	∞	∞	∞	6

Note: ∞ indicates infeasible state/action combination

Example evaluation of a trial policy

Consider the inventory replenishment policy (s,S)=(1,5), i.e., if inventory position is £ the reorder point 1, order enough to raise the inventory level to 5.

State	action
BO= three	SOH= 5
BO= two	SOH= 5
BO= one	SOH= 5
SOH= zero	SOH= 5
SOH= one	SOH= 5
SOH= two	SOH= 2
SOH= three	SOH= 3
SOH= four	SOH= 4
SOH= five	SOH= 5
SOH= six	SOH= 6

Let's use as a criterion the average cost per day in steady state.
The trial policy defines a Markov chain model.

Transition Probability Matrix for this policy:

	1	2	3	4	5	6	7	8	9	10
1	0.0119	0.0216	0.0504	0.101	0.168	0.224	0.224	0.149	0.0498	0
2	0.0119	0.0216	0.0504	0.101	0.168	0.224	0.224	0.149	0.0498	0
3	0.0119	0.0216	0.0504	0.101	0.168	0.224	0.224	0.149	0.0498	0
4	0.0119	0.0216	0.0504	0.101	0.168	0.224	0.224	0.149	0.0498	0
5	0.0119	0.0216	0.0504	0.101	0.168	0.224	0.224	0.149	0.0498	0
6	0.185	0.168	0.224	0.224	0.149	0.0498	0	0	0	0
7	0.0839	0.101	0.168	0.224	0.224	0.149	0.0498	0	0	0
8	0.0335	0.0504	0.101	0.168	0.224	0.224	0.149	0.0498	0	0
9	0.0119	0.0216	0.0504	0.101	0.168	0.224	0.224	0.149	0.0498	0
10	0.0038	0.0081	0.0216	0.0504	0.101	0.168	0.224	0.224	0.149	0.0498

i	State	Pi	C	Pi×C
1	BO= three	0.056	55	3.08
2	BO= two	0.0627	30	1.88
3	BO= one	0.104	15	1.56
4	SOH= zero	0.148	10	1.48
5	SOH= one	0.178	11	1.96
6	SOH= two	0.181	2	0.362
7	SOH= three	0.15	3	0.451
8	SOH= four	0.0908	4	0.363
9	SOH= five	0.0288	5	0.144
10	SOH= six	0	6	0

The average cost / period in steady state is $11.3

Value Iteration Method

state	Value
BO= three 1	55
BO= two 2	30
BO= one 3	15
SOH= zero 4	0
SOH= one 5	1
SOH= two 6	2
SOH= three 7	3
SOH= four 8	4
SOH= five 9	5
SOH= six 10	6

Initial Values

state	Value
BO= three 1	58.82697
BO= two 2	33.82697
BO= one 3	18.82697
SOH= zero 4	13.82697
SOH= one 5	14.82697
SOH= two 6	15.82697
SOH= three 7	13.83265
SOH= four 8	10.18689
SOH= five 9	9.19363
SOH= six 10	9.82697

Iteration #1

Max & Min ΔV ={13.8,3.83}, gap=261%

state	Value
BO= three 1	67.9996
BO= two 2	42.9996
BO= one 3	27.9996
SOH= zero 4	22.9996
SOH= one 5	23.9996
SOH= two 6	24.9996
SOH= three 7	23.9829
SOH= four 8	21.3389
SOH= five 9	19.8898
SOH= six 10	18.9996

Iteration #2

Max & Min ΔV ={11.15,9.17},gap=21.6%

state	Value
BO= three 1	78.0623
BO= two 2	53.0623
BO= one 3	38.0623
SOH= zero 4	33.0623
SOH= one 5	34.0623
SOH= two 6	35.0623
SOH= three 7	33.2041
SOH= four 8	30.7561
SOH= five 9	29.6530
SOH= six 10	29.0623

Iteration #3

Max & Min ΔV ={10.06,9.22},gap=9.12%

iteration	Max ΔV	Min ΔV	gap (%)
1	1.38270E1	3.82697E0	2.61303E2
2	1.11520E1	9.17262E0	2.15792E1
3	1.00627E1	9.22129E0	9.12428E0
4	1.00208E1	9.68481E0	3.46902E0
5	9.82103E0	9.70153E0	1.23176E0
6	9.81508E0	9.77382E0	4.22165E-1
7	9.79001E0	9.77588E0	1.44524E-1
8	9.78930E0	9.78448E0	4.93237E-2
9	9.78636E0	9.78472E0	1.68360E-2
10	9.78628E0	9.78572E0	5.74510E-3
11	9.78594E0	9.78575E0	1.96053E-3
12	9.78593E0	9.78586E0	6.69020E-4
13	9.78589E0	9.78587E0	2.28301E-4
14	9.78589E0	9.78588E0	7.79065E-5

Converged!

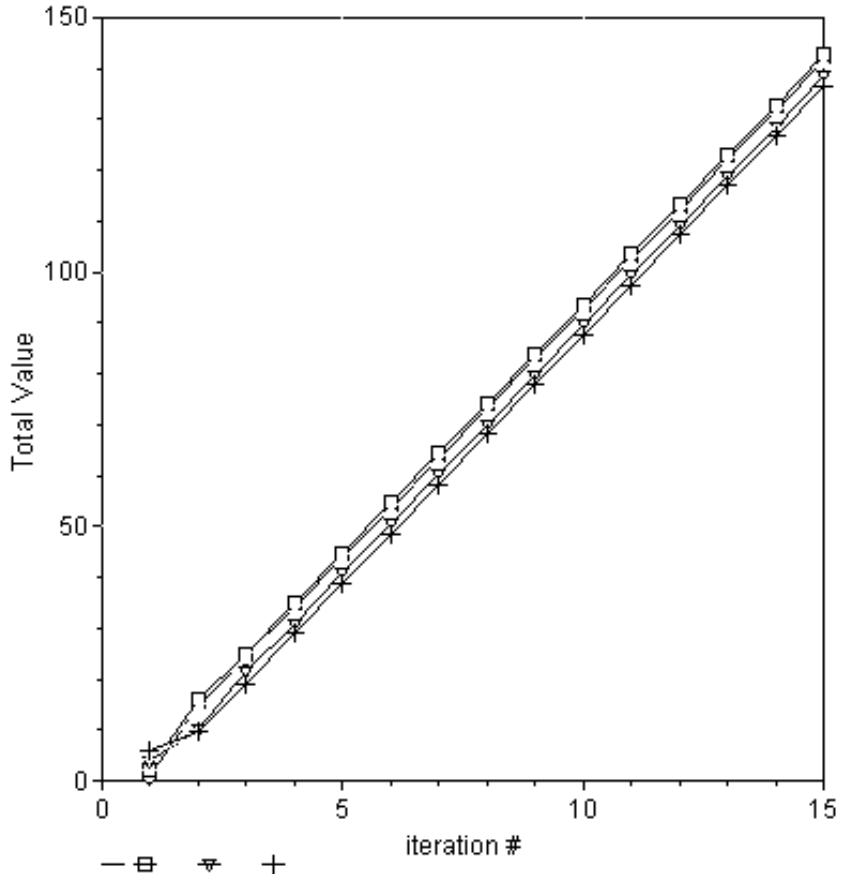
Optimal policy:

Average cost/period

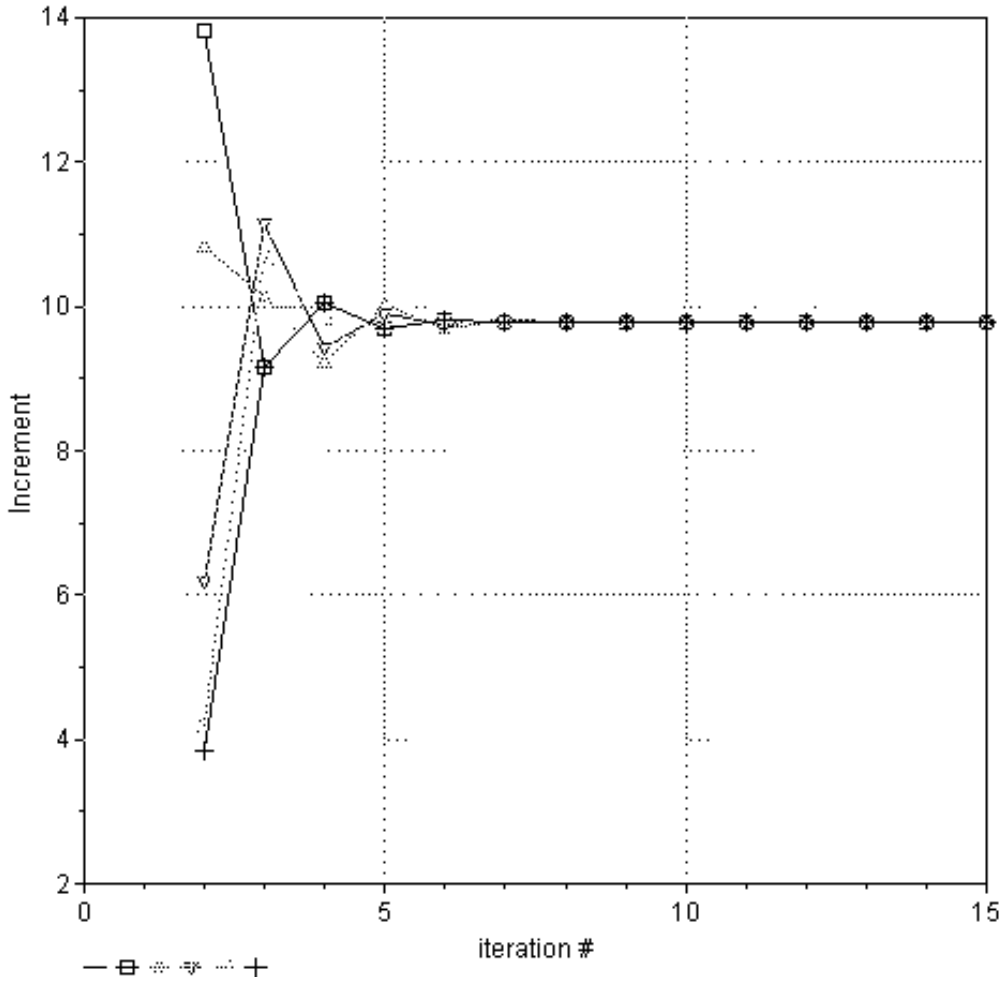
State	Action	V
BO= three	SOH= 6	9.78588
BO= two	SOH= 6	9.78588
BO= one	SOH= 6	9.78588
SOH= zero	SOH= 6	9.78588
SOH= one	SOH= 6	9.78588
SOH= two	SOH= 6	9.78588
SOH= three	SOH= 3	9.78589
SOH= four	SOH= 4	9.78589
SOH= five	SOH= 5	9.78588
SOH= six	SOH= 6	9.78588

Optimal policy is (s,S) = (1,6)!

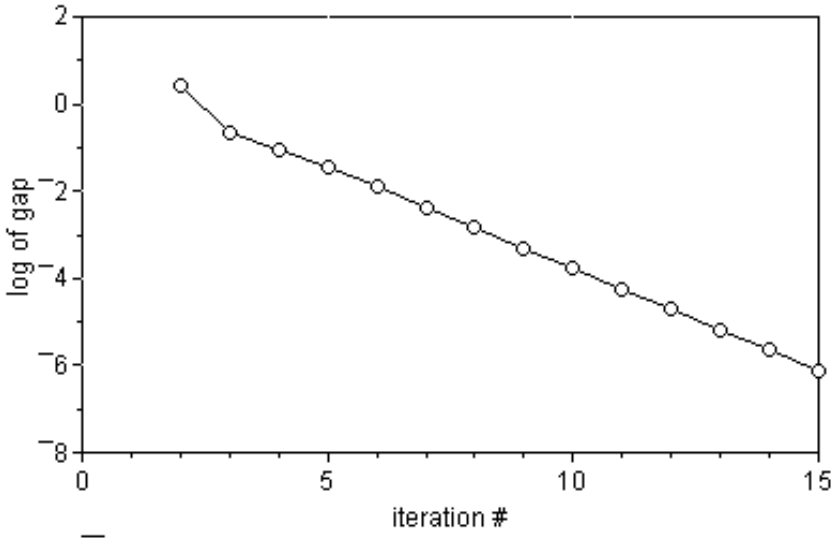
Plot of V(i)



Plot of increments ΔV (one per state)



Log (base 10) of gap between max & min ΔV



Linear Programming Method

Maximize $\sum_i \sum_k C_i^k X_i^k$

$$\sum_k X_j^k = \sum_i \sum_k p_{ij}^k X_i^k \qquad \forall j$$

$$\sum_i \sum_k X_i^k = 1, \quad X_i^k \geq 0 \forall i \,\&\, k$$

Tableau: (i~state, k~action)

k:	1	2	3	4	5	6	7	1	2
i:	1	1	1	1	1	1	1	2	2
Min	55	55	55	55	55	55	55	30	30
	0.423	0.647	0.815	0.916	0.966	0.988	0.996	-0.577	-0.353
	-0.224	-0.224	-0.168	-0.101	-0.0504	-0.0216	-0.0081	0.776	0.776
	-0.149	-0.224	-0.224	-0.168	-0.101	-0.0504	-0.0216	-0.149	-0.224
	-0.0498	-0.149	-0.224	-0.224	-0.168	-0.101	-0.0504	-0.0498	-0.149
	0	-0.0498	-0.149	-0.224	-0.224	-0.168	-0.101	0	-0.0498
	0	0	-0.0498	-0.149	-0.224	-0.224	-0.168	0	0
	0	0	0	-0.0498	-0.149	-0.224	-0.224	0	0
	0	0	0	0	-0.0498	-0.149	-0.224	0	0
	0	0	0	0	0	-0.0498	-0.149	0	0
	1	1	1	1	1	1	1	1	1

k=	3	4	5	6	7	1	2	3	4
i=	2	2	2	2	2	3	3	3	3
	30	30	30	30	30	15	15	15	15
	-0.185	-0.0839	-0.0335	-0.0119	-0.0038	-0.577	-0.353	-0.185	-0.0839
	0.832	0.899	0.95	0.978	0.992	-0.224	-0.224	-0.168	-0.101
	-0.224	-0.168	-0.101	-0.0504	-0.0216	0.851	0.776	0.776	0.832
	-0.224	-0.224	-0.168	-0.101	-0.0504	-0.0498	-0.149	-0.224	-0.224
	-0.149	-0.224	-0.224	-0.168	-0.101	0	-0.0498	-0.149	-0.224
	-0.0498	-0.149	-0.224	-0.224	-0.168	0	0	-0.0498	-0.149
	0	-0.0498	-0.149	-0.224	-0.224	0	0	0	-0.0498
	0	0	-0.0498	-0.149	-0.224	0	0	0	0
	0	0	0	-0.0498	-0.149	0	0	0	0
	1	1	1	1	1	1	1	1	1

k=	5	6	7	1	2	3	4	5	6	7
i=	3	3	3	4	4	4	4	4	4	4
	15	15	15	0	10	10	10	10	10	10
	-0.0335	-0.0119	-0.0038	-0.577	-0.353	-0.185	-0.0839	-0.0335	-0.0119	-0.0038
	-0.0504	-0.0216	-0.0081	-0.224	-0.224	-0.168	-0.101	-0.0504	-0.0216	-0.0081
	0.899	0.95	0.978	-0.149	-0.224	-0.224	-0.168	-0.101	-0.0504	-0.0216
	-0.168	-0.101	-0.0504	0.95	0.851	0.776	0.776	0.832	0.899	0.95
	-0.224	-0.168	-0.101	0	-0.0498	-0.149	-0.224	-0.224	-0.168	-0.101
	-0.224	-0.224	-0.168	0	0	-0.0498	-0.149	-0.224	-0.224	-0.168
	-0.149	-0.224	-0.224	0	0	0	-0.0498	-0.149	-0.224	-0.224
	-0.0498	-0.149	-0.224	0	0	0	0	-0.0498	-0.149	-0.224
	0	-0.0498	-0.149	0	0	0	0	0	-0.0498	-0.149
	1	1	1	1	1	1	1	1	1	1

k=	2	3	4	5	6	7	3	4	5	6
i=	5	5	5	5	5	5	6	6	6	6
	1	11	11	11	11	11	2	12	12	12
	-0.353	-0.185	-0.0839	-0.0335	-0.0119	-0.0038	-0.185	-0.0839	-0.0335	-0.0119
	-0.224	-0.168	-0.101	-0.0504	-0.0216	-0.0081	-0.168	-0.101	-0.0504	-0.0216
	-0.224	-0.224	-0.168	-0.101	-0.0504	-0.0216	-0.224	-0.168	-0.101	-0.0504
	-0.149	-0.224	-0.224	-0.168	-0.101	-0.0504	-0.224	-0.224	-0.168	-0.101
	0.95	0.851	0.776	0.776	0.832	0.899	-0.149	-0.224	-0.224	-0.168
	0	-0.0498	-0.149	-0.224	-0.224	-0.168	0.95	0.851	0.776	0.776
	0	0	-0.0498	-0.149	-0.224	-0.224	0	-0.0498	-0.149	-0.224
	0	0	0	-0.0498	-0.149	-0.224	0	0	-0.0498	-0.149
	0	0	0	0	-0.0498	-0.149	0	0	0	-0.0498
	1	1	1	1	1	1	1	1	1	1

k=	7	4	5	6	7	5	6	7	6
i=	6	7	7	7	7	8	8	8	9
	12	3	13	13	13	4	14	14	5
	-0.0038	-0.0839	-0.0335	-0.0119	-0.0038	-0.0335	-0.0119	-0.0038	-0.0119
	-0.0081	-0.101	-0.0504	-0.0216	-0.0081	-0.0504	-0.0216	-0.0081	-0.0216
	-0.0216	-0.168	-0.101	-0.0504	-0.0216	-0.101	-0.0504	-0.0216	-0.0504
	-0.0504	-0.224	-0.168	-0.101	-0.0504	-0.168	-0.101	-0.0504	-0.101
	-0.101	-0.224	-0.224	-0.168	-0.101	-0.224	-0.168	-0.101	-0.168
	0.832	-0.149	-0.224	-0.224	-0.168	-0.224	-0.224	-0.168	-0.224
	-0.224	0.95	0.851	0.776	0.776	-0.149	-0.224	-0.224	-0.224
	-0.224	0	-0.0498	-0.149	-0.224	0.95	0.851	0.776	-0.149
	-0.149	0	0	-0.0498	-0.149	0	-0.0498	-0.149	0.95
	1	1	1	1	1	1	1	1	1

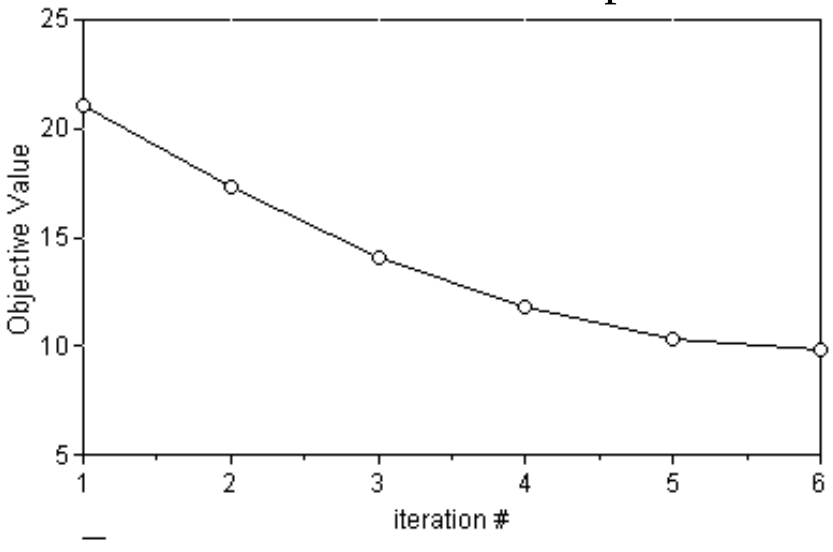
k=	7	7	|	R
i=	9	10	|	H
	15	6	|	S
	-0.0038	-0.0038	|	0
	-0.0081	-0.0081	|	0
	-0.0216	-0.0216	|	0
	-0.0504	-0.0504	|	0
	-0.101	-0.101	|	0
	-0.168	-0.168	|	0
	-0.224	-0.224	|	0
	-0.224	-0.224	|	0
	0.851	-0.149	|	0
	1	1	|	1

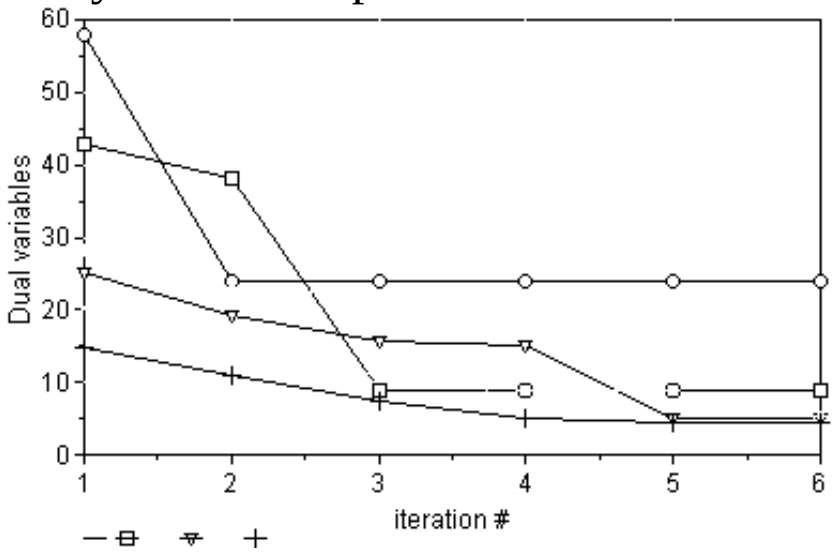
Initial policy found by Phase I:

State		Action	
1)	BO= three	7)	SOH= 6
2)	BO= two	1)	SOH= 0
3)	BO= one	1)	SOH= 0
4)	SOH= zero	1)	SOH= 0
5)	SOH= one	2)	SOH= 1
6)	SOH= two	3)	SOH= 2
7)	SOH= three	4)	SOH= 3
8)	SOH= four	5)	SOH= 4
9)	SOH= five	6)	SOH= 5
10)	SOH= six	7)	SOH= 6

Initial policy is of type (s,S) = (-3,6)

Six iterations of the simplex method yield the optimal solution:





	State	Action		P{i}	R{i}
1)	BO= three	7)	SOH= 6	0.0235265	49
2)	BO= two	7)	SOH= 6	0.0324952	24
3)	BO= one	7)	SOH= 6	0.062673	9
4)	SOH= zero	7)	SOH= 6	0.104148	4
5)	SOH= one	7)	SOH= 6	0.14779	5
6)	SOH= two	7)	SOH= 6	0.178159	6
7)	SOH= three	4)	SOH= 3	0.181227	4.38909
8)	SOH= four	5)	SOH= 4	0.150444	1.85787
9)	SOH= five	6)	SOH= 5	0.0907828	0.650524
10)	SOH= six	7)	SOH= 6	0.0287543	9.78588

Average cost/stage = 9.78588

Note that the optimal policy is of type **(s, S)** = (2, 6)!

Policy Iteration Method

We must first choose an **initial policy**.

Arbitrarily, let's choose the policy

> "Order enough to meet the next day's expected demand (i.e., action "SOH=3") only if the stock on hand is less than 3".

The next step is "value determination" which evaluates this policy:

Value Determination

Value Determination Step:

	State		Action	
1	BO= three	4	SOH=	3
2	BO= two	4	SOH=	3
3	BO= one	4	SOH=	3
4	SOH= zero	4	SOH=	3
5	SOH= one	4	SOH=	3
6	SOH= two	4	SOH=	3
7	SOH= three	4	SOH=	3
8	SOH= four	5	SOH=	4
9	SOH= five	6	SOH=	5
10	SOH= six	7	SOH=	6

i	Vi
1	61.7939
2	36.7939
3	21.7939
4	16.7939
5	17.7939
6	18.7939
7	9.79394
8	6.10125
9	2.96913
10	0

Average cost/period = g(R) = **16.8071**

Policy Improvement

We now determine the effect on the cost if we were to modify the policy by

- choosing an alternative action in the ***first*** stage, and then
- following the current policy thereafter

This is equivalent to "*pricing*", i.e., the computation of the *reduced costs* in the simplex method.

Policy Improvement Step: Evaluation of alternate actions

C'[k] = cost if action k is selected for one stage

ΔC[k] = improvement (if <0)

State #1, BO= three

Current Policy: action #4:SOH=3

k	action	C'	ΔC
1	SOH= 0	102.978	24.377
2	SOH= 1	93.3193	14.7183
3	SOH= 2	84.8369	6.23583
4	SOH= 3	78.601	0
5	SOH= 4	73.9083	-4.69269
6	SOH= 5	69.7762	-8.82481
7	SOH= 6	65.8071	-12.7939

State #2, BO= two

Current Policy: action #4:SOH=3

k	action	C'	ΔC
1	SOH= 0	77.978	24.377
2	SOH= 1	68.3193	14.7183
3	SOH= 2	59.8369	6.23583
4	SOH= 3	53.601	0
5	SOH= 4	48.9083	-4.69269
6	SOH= 5	44.7762	-8.82481
7	SOH= 6	40.8071	-12.7939

State #3, BO= one

Current Policy: action #4: SOH=3

k	action	C'	ΔC
1	SOH= 0	62.978	24.377
2	SOH= 1	53.3193	14.7183
3	SOH= 2	44.8369	6.23583
4	SOH= 3	38.601	0
5	SOH= 4	33.9083	-4.69269
6	SOH= 5	29.7762	-8.82481
7	SOH= 6	25.8071	-12.7939

State #4, SOH= zero

Current Policy: action #4: SOH=3

k	action	C'	ΔC
1	SOH= 0	47.978	14.377
2	SOH= 1	48.3193	14.7183
3	SOH= 2	39.8369	6.23583
4	SOH= 3	33.601	0
5	SOH= 4	28.9083	-4.69269
6	SOH= 5	24.7762	-8.82481
7	SOH= 6	20.8071	-12.7939

State #5, SOH= one

Current Policy: action #4:SOH=3

k	action	C'	ΔC
1	SOH= 1	39.3193	4.7183
2	SOH= 2	40.8369	6.23583
3	SOH= 3	34.601	0
4	SOH= 4	29.9083	-4.69269
5	SOH= 5	25.7762	-8.82481
6	SOH= 6	21.8071	-12.7939

State #6, SOH= two

Current Policy: action #4:SOH=3

k	action	C'	ΔC
1	SOH= 2	31.8369	-3.76417
2	SOH= 3	35.601	0
3	SOH= 4	30.9083	-4.69269
4	SOH= 5	26.7762	-8.82481
5	SOH= 6	22.8071	-12.7939

State #7, SOH= three

Current Policy: action #4:SOH=3

k	action	C'	ΔC
1	SOH= 3	26.601	0
2	SOH= 4	31.9083	5.30731
3	SOH= 5	27.7762	1.17519
4	SOH= 6	23.8071	-2.79394

State #8, SOH= four

Current Policy: action #5:SOH=4

k	action	C'	ΔC
1	SOH= 4	22.9083	0
2	SOH= 5	28.7762	5.86788
3	SOH= 6	24.8071	1.89875

State #9, SOH= five

Current Policy: action #6:SOH=5

k	action	C'	ΔC
1	SOH= 5	19.7762	0
2	SOH= 6	25.8071	6.03087

State #10, SOH= six

Current Policy: action #7, SOH=6

k	action	C'	ΔC
1	SOH= 6	16.8071	0

Several changes, any one of which would result in improvements, are indicated by **ΔC < 0**:

State #1, BO= three

action	ΔC
SOH= 4	-4.69269
SOH= 5	-8.82481
SOH= 6	-12.7939

State #2, BO= two

action	ΔC
SOH= 4	-4.69269
SOH= 5	-8.82481
SOH= 6	-12.7939

State #3, BO= one

action	ΔC
SOH= 4	-4.69269
SOH= 5	-8.82481
SOH= 6	-12.7939

State #4, SOH= zero

action	ΔC
SOH= 4	-4.69269
SOH= 5	-8.82481
SOH= 6	-12.7939

State #5, SOH= one

action	ΔC
SOH= 4	-4.69269
SOH= 5	-8.82481
SOH= 6	-12.7939

State #6, SOH= two

action	ΔC
SOH= 2	-3.76417
SOH= 4	-4.69269
SOH= 5	-8.82481
SOH= 6	-12.7939

State #7, SOH= three

action	ΔC
SOH= 6	-2.79394

If we were to revise the current policy by selecting a different action for a ***single*** state, the result would be equivalent to a ***pivot*** to change the current basis in the simplex method.

Revising the current policy by simultaneously changing the action for ***several*** states is equivalent to "***block pivoting***" in the simplex method.

Selection of Actions

In the simplex method, a common "rule of thumb" is to select the column having the "*most negative*" (i.e., lowest negative) reduced cost to enter the basis, although this does not guarantee the shortest sequence of pivots to the optimum.

Likewise, in the policy improvement algorithm, for each state it is usual to select the action with the "most negative" value of ΔC to replace the current action, although any action having $\mathbf{\Delta C < 0}$ will suffice.

Partial Pricing

When solving large LP problems, it is common in the simplex method to use "*partial pricing*", that is, not necessarily computing the reduced cost of every variable before selecting one having negative reduced cost to enter the basis.

Likewise, in the policy improvement algorithm, one need not compute ΔC for every alternative action if one has already been found with $\mathbf{\Delta C < 0.}$

Revising the Policy

Suppose that we replace the action in each state with the action having smallest value of $\Delta C<0$, so that the new policy is

State		Action	
1	BO= three	7	SOH= 6
2	BO= two	7	SOH= 6
3	BO= one	7	SOH= 6
4	SOH= zero	7	SOH= 6
5	SOH= one	7	SOH= 6
6	SOH= two	7	SOH= 6
7	SOH= three	7	SOH= 6
8	SOH= four	5	SOH= 4
9	SOH= five	6	SOH= 5
10	SOH= six	7	SOH= 6

*If Stock-on-Hand is **3** or fewer, order enough so that the new Stock-on-Hand is **6**, i.e.,*
*a policy of type **(s,S)= (3,6)***

Value Determination

We must now evaluate this new policy:

	State		Action		i	Vi
1	BO= three	7	SOH=	6	1	49
2	BO= two	7	SOH=	6	2	24
3	BO= one	7	SOH=	6	3	9
4	SOH= zero	7	SOH=	6	4	4
5	SOH= one	7	SOH=	6	5	5
6	SOH= two	7	SOH=	6	6	6
7	SOH= three	7	SOH=	6	7	7
8	SOH= four	5	SOH=	4	8	1.69038
9	SOH= five	6	SOH=	5	9	0.6619
10	SOH= six	7	SOH=	6	10	0

Average cost/period = g(R) =**10.335**

Policy Improvement

Policy Improvement Step:
Evaluation of alternate actions

State #1, BO= three
Current Policy: action #7:SOH= 6

k	action	C'	ΔC
1	SOH= 0	90.1841	30.8491
2	SOH= 1	80.5254	21.1904
3	SOH= 2	72.0429	12.7079
4	SOH= 3	66.305	6.96996
5	SOH= 4	63.0254	3.69038
6	SOH= 5	60.9969	1.6619

State #2, BO= two
Current Policy: action #7:SOH= 6

k	action	C'	ΔC
1	SOH= 0	65.1841	30.8491
2	SOH= 1	55.5254	21.1904
3	SOH= 2	47.0429	12.7079
4	SOH= 3	41.305	6.96996
5	SOH= 4	38.0254	3.69038
6	SOH= 5	35.9969	1.6619

State #3, BO= one
Current Policy: action #:SOH= 6

k	action	C'	ΔC
1	SOH= 0	50.1841	30.8491
2	SOH= 1	40.5254	21.1904
3	SOH= 2	32.0429	12.7079
4	SOH= 3	26.305	6.96996
5	SOH= 4	23.0254	3.69038
6	SOH= 5	20.9969	1.6619

State #4, SOH= zero
Current Policy: action #:SOH= 6

k	action	C'	ΔC
1	SOH= 0	35.1841	20.8491
2	SOH= 1	35.5254	21.1904
3	SOH= 2	27.0429	12.7079
4	SOH= 3	21.305	6.96996
5	SOH= 4	18.0254	3.69038
6	SOH= 5	15.9969	1.6619

State #5, SOH= one

Current Policy: action #7:SOH= 6

k	action	C'	ΔC
1	SOH= 1	26.5254	11.1904
2	SOH= 2	28.0429	12.7079
3	SOH= 3	22.305	6.96996
4	SOH= 4	19.0254	3.69038
5	SOH= 5	16.9969	1.6619

State #6, SOH= two

Current Policy: action #7:SOH= 6

k	action	C'	ΔC
1	SOH= 2	19.0429	2.70792
2	SOH= 3	23.305	6.96996
3	SOH= 4	20.0254	3.69038
4	SOH= 5	17.9969	1.6619

State #7, SOH= three

Current Policy: action #7:SOH= 6

k	action	C'	ΔC
1	SOH= 3	14.305	3.03004
2	SOH= 4	21.0254	3.69038
3	SOH= 5	18.9969	1.6619

State #8, SOH= four

Current Policy: action #5:SOH= 4

k	action	C'	ΔC
2	SOH= 5	19.9969	7.97152
3	SOH= 6	18.335	6.30962

State #9, SOH= five

Current Policy: action #6:SOH= 5

k	action	C'	ΔC
2	SOH= 6	19.335	8.3381

Revising the Policy

Only one improving change is indicated, namely, changing the action in the state "SOH=3" from "SOH=6" to "SOH=3", that is, leaving the inventory level unchanged instead of increasing it to 6 by ordering three additional units.

Value Determination

Value Determination Results

	State	Action		i	Vi
1	BO= three	7	SOH= 6	1	49
2	BO= two	7	SOH= 6	2	24
3	BO= one	7	SOH= 6	3	9
4	SOH= zero	7	SOH= 6	4	4
5	SOH= one	7	SOH= 6	5	5
6	SOH= two	7	SOH= 6	6	6
7	SOH= three	4	SOH= 3	7	4.38909
8	SOH= four	5	SOH= 4	8	1.85787
9	SOH= five	6	SOH= 5	9	0.650524
10	SOH= six	7	SOH= 6	10	0

Average cost/period = g(R) = **9.78588**

Policy Improvement

State #1, BO= three

Current Policy: action #7:SOH= 6

k	action	C'	ΔC
1	SOH= 0	90.1841	31.3982
2	SOH= 1	80.5254	21.7395
3	SOH= 2	72.0429	13.257
4	SOH= 3	66.175	7.38909
5	SOH= 4	62.6438	3.85787
6	SOH= 5	60.4364	1.65052

State #2, BO = two

Current Policy: action #7:SOH= 6

k	action	C'	ΔC
1	SOH= 0	65.1841	31.3982
2	SOH= 1	55.5254	21.7395
3	SOH= 2	47.0429	13.257
4	SOH= 3	41.175	7.38909
5	SOH= 4	37.6438	3.85787
6	SOH= 5	35.4364	1.65052

State #3, BO= one

Current Policy: action #7:SOH= 6

k	action	C'	ΔC
1	SOH= 0	50.1841	31.3982
2	SOH= 1	40.5254	21.7395
3	SOH= 2	32.0429	13.257
4	SOH= 3	26.175	7.38909
5	SOH= 4	22.6438	3.85787
6	SOH= 5	20.4364	1.65052

State #4, SOH= zero

Current Policy: action #7:SOH= 6

k	action	C'	ΔC
1	SOH= 0	35.1841	21.3982
2	SOH= 1	35.5254	21.7395
3	SOH= 2	27.0429	13.257
4	SOH= 3	21.175	7.38909
5	SOH= 4	17.6438	3.85787
6	SOH= 5	15.4364	1.65052

State #5, SOH= one

Current Policy: action #7:SOH= 6

k	action	C'	ΔC
1	SOH= 1	26.5254	11.7395
2	SOH= 2	28.0429	13.257
3	SOH= 3	22.175	7.38909
4	SOH= 4	18.6438	3.85787
5	SOH= 5	16.4364	1.65052

State #6, SOH= two

Current Policy: action #7:SOH= 6

k	action	C'	ΔC
1	SOH= 2	19.0429	3.25704
2	SOH= 3	23.175	7.38909
3	SOH= 4	19.6438	3.85787
4	SOH= 5	17.4364	1.65052

State #7, SOH= three

Current Policy: action #4:SOH= 3

k	action	C'	ΔC
2	SOH= 4	20.6438	6.46878
3	SOH= 5	18.4364	4.26143
4	SOH= 6	16.7859	2.61091

State #8, SOH= four

Current Policy: action #5:SOH= 4

k	action	C'	ΔC
2	SOH= 5	19.4364	7.79265
3	SOH= 6	17.7859	6.14213

State #9, SOH= five

Current Policy: action #6:SOH= 5

k	action	C'	ΔC
2	SOH= 6	18.7859	8.34948

Since $\Delta \mathbf{C} \geq \mathbf{0}$ in for all policy changes,

the current policy is **optimal**!

(Note that the optimal policy is of type **(s,S) = (2,6)**