MDP Model of Inventory Replenishment

©Dennis Bricker, 2001
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

Steady State Distribution

The average cost / period in steady state is $11.3

Value Iteration Method

Initial Values

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

Iteration #1

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

Max & Min ΔV ={13.8,3.83}, gap=261%

Iteration #2

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

Max & Min ΔV={11.15,9.17},gap=21.6%

Iteration #3

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

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!

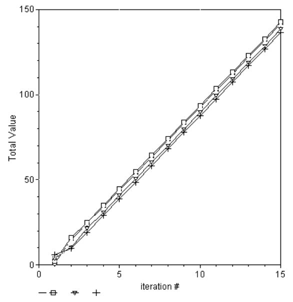
Optimal policy:

Average cost/period

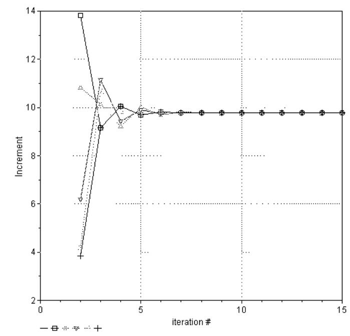
State	Action	V
BO= three	SOH= 6	9.78588
BO= two	SOH= 6	9.78588
BO= one	SOH= 6	9.78588
SOH= zero	SOH= 6	9.78588
SOH= one	SOH= 6	9.78588
SOH= two	SOH= 6	9.78588
SOH= three	SOH= 3	9.78589
SOH= four	SOH= 4	9.78589
SOH= five	SOH= 5	9.78588
SOH= six	SOH= 6	9.78588

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

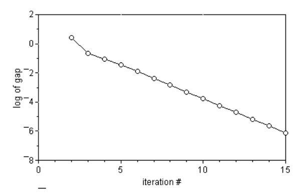
Plot of V(i)



Plot of increments ΔV (one per state)



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



Linear Programming Method

$$\text{Maximize} \quad \sum_i \sum_k C_i^k X_i^k$$

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

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

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

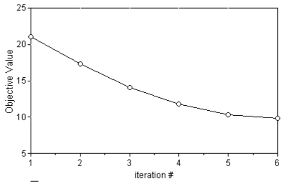
Linear Programming Tableau (Average Cost Criterion)

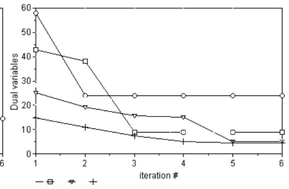
Initial policy found by Phase I:

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

Optimal policy is of type (s, S) = (2, 6)!

Policy Iteration Method