

```

1 import numpy
2 from binomialPoisson import *
3
4 hugeNumber = float("inf")
5 unused = -1000
6
7 stages = 10
8 maxMachines = 25
9 chanceFailure = 0.05
10
11 machinesNeeded = numpy.array([unused,10, 12, 5, 9, 15, 16, 18, 20, 21, 17])
12
13 machineCost = 20000.00
14 shortageCost = 5000.00
15 salvageValue = 13000.00
16
17 initialState = 11 # start with 11 machines
18
19 discountRate = 0.01
20
21 # End of input data
22
23 beta = 1/(1 + discountRate)
24
25 # Precomputes probability distributions so that
26 # failureProb[i,k] is the chance that if you have i machines, k will fail
27 failureProb = numpy.zeros([maxMachines + 1, maxMachines + 1])
28 for i in range(maxMachines+1) :
29     failureProb[i,0:i+1] = binomial(i,chanceFailure)
30
31 f = numpy.zeros([stages + 2, maxMachines + 1])
32 x = numpy.zeros([stages + 1, maxMachines + 1], dtype=int)
33
34 # We do this as a cost problem, so salvage value is negative
35 for i in range(maxMachines + 1) :
36     f[stages+1][i] = -salvageValue*i
37
38 for t in range(stages,0,-1) :
39
40     for i in range(maxMachines+1) :
41
42         value = hugeNumber
43         maxPurchase = maxMachines - i
44
45         for p in range(maxPurchase+1):
46
47             moveValue = p*machineCost
48
49             for e in range(i + 1) : # Here, e is number of failures
50                 working = i - e
51                 machinesShort = max(0,machinesNeeded[t] - working)
52                 shortageCharge = shortageCost*machinesShort
53                 j = working + p
54                 moveValue += failureProb[i,e]*(shortageCharge + beta*f[t+1,j])
55
56             if moveValue < value :
57                 value = moveValue
58                 bestMove = p
59
60         # End of d Loop
61
62         f[t,i] = value
63         x[t,i] = bestMove
64
65     # End of i Loop
66
67 # End of t Loop
68

```

```
69 print("Optimal cost is " + str(f[1,initialState]))
70 print("Stage 1: purchase " + str(x[1,initialState]) + " machines")
71 for t in range(2, stages + 1) :
72     print("At stage " + str(t) + " : ")
73     sumOfXt = sum(x[t,:])
74     for i in range(maxMachines + 1) :
75         if sumOfXt == 0 :
76             print("    if have " + str(i) + " or more machines, purchase 0")
77             break
78     print("    if have " + str(i) + " machines, purchase " + str(x[t,i]))
79     sumOfXt -= x[t, i]
80
```