

```

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 f = numpy.zeros([stages + 2, maxMachines + 1])
26 x = numpy.zeros([stages + 1, maxMachines + 1], dtype=int)
27
28 # We do this as a cost problem, so salvage value is negative
29 for i in range(maxMachines + 1) :
30     f[stages+1][i] = -salvageValue*i
31
32 for t in range(stages,0,-1) :
33
34     for i in range(maxMachines+1) :
35
36         # The program would run a bit faster than this if we were to
37         # precompute and store these values in a two-dimensional array
38         # indexed by i and the number of failures
39         failureDist = binomial(i,chanceFailure)
40
41         value = hugeNumber
42         maxPurchase = maxMachines - i
43
44         for p in range(maxPurchase+1):
45
46             moveValue = p*machineCost
47
48             for e in range(i + 1) :         # Here, e is number of failures
49                 working = i - e
50                 machinesShort = max(0,machinesNeeded[t] - working)
51                 shortageCharge = shortageCost*machinesShort
52                 j = working + p
53                 moveValue += failureDist[e]*(shortageCharge + beta*f[t+1,j])
54
55             if moveValue < value :
56                 value = moveValue
57                 bestMove = p
58
59         # End of d Loop
60
61         f[t,i] = value
62         x[t,i] = bestMove
63
64     # End of i Loop
65
66 # End of t Loop
67
68 print("Optimal cost is " + str(f[1,initialState]))

```

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