Lecture 12: Feb 28, 2013

In this lecture, we wrap-up our inventory planning example and introduce how we can use AMPL to help us carry out dynamic programming computation.

1 Recap: Solving inventory planning problems using DP

Recall our inventory planning problem:

- Input:
  - T = number of periods in the "planning horizon"
  - For each period $k \in \{1, 2, \ldots, T\}$, there is a demand d_k, a per-unit production cost c_k, a fixed cost of production F_k, and a per-unit holding cost h_k.
- Decision to make: to determine the quantity to be produced in each period
- Constraint: demand in each period must be satisfied by the end of the period
- Objective: to minimize total cost

Period (k)	d_k	c_k	F_k	h_k
1	10	3	5	0.2
2	40	2	20	0.3
3	20	4	10	0.5
4	50	3	10	0.8

Also assume that the production quantities can only be in multiples of 10.

1.1 Dynamic programming formulation

1.1.1 Specify the stages

Stage k corresponds to period k. So, there will be five stages: $1, 2, 3, 4, 5$, where the fifth stage is a dummy stage, indicating the end of the fourth period.

1.1.2 Specify the states

The state I corresponds to inventory level I at the beginning of the period.

$$S_k = \{0, 10, 20, \ldots, 120\}.$$

1.1.3 Specify the sets of allowable decisions at each state in each stage

Let x_k denote the quantity to produce at stage k. If we start with an inventory of I units at the beginning of stage k, then we want to make sure that:

- We have enough items to satisfy demand: $I + x_k \geq d_k$

- And since we assume that our inventory level is at most 120, we need only consider x_k such that
$$I + x_k - d_k \leq 120.$$
- So, in summary, we formulate the set of allowable decisions at state I in stage k as:
$$Q_{k,I} = \{x \in \{0, 10, \dots\} \mid I + x_k \geq d_k \text{ and } I + x_k - d_k \leq 120\}.$$

1.1.4 Describe in words the optimization function to be solved at each state in each stage

Consider a state I in stage k. That is, the inventory level at the beginning of period k is I. Then,

$$f_k^*(I) \quad = \quad \text{the minimum total cost to meet demands in period } k \text{ until period } T+1$$

1.1.5 Specify boundary conditions

$$f_5^*(I) = 0, \ \forall I \in \{0, 10, \dots, 120\}.$$

1.1.6 Write the recurrence relation for the optimization function described in step 4

$$f_k^*(I) = \min_{x_k \in Q_{k,I}} \left\{ \underbrace{(x_k * c_k + \mathbb{1}_{x_k > 0} F_k + (I + x_k - d_k) * h_k)}_{\text{cost due to producing } x_k \text{ in period } k} + f_{k+1}^*(I + x_k - d_k) \right\}.$$

1.1.7 Compute the value of $f_k^*(I)$ for each state $I \in S_k$, for each stage k

To compute $f_k^*(I)$, we only need to use the recurrence relation we described in step 6. In order to do the computation, since each f_k^* refer to a value f_{k+1}^*, we do the computation starting from $k = T + 1$ (the boundary conditions), then $k = T, T-1, \dots, 1$.

Within each stage k, we solve for $f_k^*(I)$ for all states $I \in S_k$, before proceeding to stage $k-1$. That is, we complete the following table starting from each row in the leftmost column, proceeding to the next left-most column, etc.

Possible States (i)	Stage 5	Stage 4		Stage 3		Stage 2		Stage 1	
	f*_5(i)	f*_4(i)	x*_4	f*_3(i)	x*_3	f*_2(i)	x*_2	f*_1(i)	x*_1
0	0	160	50	250	20	286	110	321	10
10	0	130	40	210	10	266	100		
20	0	100	30	160	0	246	90		
30	0	70	20	135	0	226	80		
40	0	40	10	110	0	206	70		
50	0	0	0	85	0	186	60		
60	0	8	0	60	0	166	0 or 50		
70	0	16	0	25	0	144	0		
80	0	24	0	38	0	122	0		
90	0	32	0	51	0	100	0		
100	0	40	0	64	0	78	0		
110	0	48	0	77	0	46	0		
120	0	56	0	90	0	62	0		

1.1.8 Trace backwards to find the optimal objective function and the optimal decision.

- We begin by recognizing that our original problem, to find the minimum-cost solution to satisfy demand from period 1 to period 5, is represented by the state $I = 0$ in stage 1. So, from the table above, our minimum cost is $f_1^*(0) = 321$. From the table, we also know that the optimal quantity to produce in period 1 is $x_1^* = 10$.
- Given that $x_1^* = 10$, the inventory level at the beginning of period 2 is $0 + x_1^* - d_1 = 0 + 10 - 10 = 0$. So, we next look at $f_2^*(0)$, to find that $x_2^* = 110$.
- Given that $x_2^* = 110$, the inventory level at the beginning of period 3 is $0 + x_2^* - d_2 = 0 + 110 - 20 = 70$. So, we next look at $f_3^*(70)$, to find that $x_3^* = 0$.
- Given that $x_3^* = 0$, the inventory level at the beginning of period 4 is $70 + x_3^* - d_3 = 70 + 0 - 20 = 50$. So, we next look at $f_4^*(50)$, to find that $x_4^* = 0$.

Hence, we can conclude that the minimum cost is 321, obtained by the following production plan:

$$x_1 = 10,\ x_2 = 110,\ x_3 = 0, x_4 = 0.$$

2 Using AMPL for DP computation

In Optimization 1 and in the network optimization part of the semester, we have been using AMPL to solve linear programming (and possibly integer programming) formulation of various optimization problems.

The components of an AMPL formulation:

Model file In the model file, we declare what our sets and parameters are (the values of which are normally supplied by a data file). We also specify what our linear programming problem is.

Data file In the data file, we supply actual values for the sets and parameters that we indicated in the model file.

Script file In the script file, we can store any sequence of commands that we might write in the `ampl` (or `sw`) console. The advantage of doing this is that we don't have to repeatedly enter the sequence of commands (when debugging, for example). Instead, we only need to enter one line:

```
include scriptfile.txt;
```

to tell AMPL to look at the file `scriptfile.txt` for the sequence of commands.

As you have noticed, there is no linear program to be solved when we're using dynamic programming. So, in the model file, we won't be specifying any linear program. That is, we won't have any AMPL decision variables (`var ...`), AMPL objective function (`maximize ...`), or AMPL constraint (`subject to ...`).

Instead, we our model file will consists of parameters and sets:

- Some of the parameters and sets will have values that are supplied by the data file
- Some other parameters and sets will be assigned values when the model is running. These parameters and sets represent our "DP Table" which we fill out in "step 7".

2.1 AMPL files for our inventory planning problem

We will start with the "easy" part, the data file. The data file consists of information on the number of periods, demands, production costs, fixed costs, and holding costs.

```
param T := 4; # number of periods

param: d c F h := # demand and costs
1   10  3  5 0.2
2   40  2 20 0.3
3   20  4 10 0.5
4   50  3 10 0.8;
```

Then, the following is the model file:

```
# Parameters specified in the data file

param T; # number of periods

param c {k in 1..T};
# cost per unit in period k

param F {k in 1..T};
# fixed cost in period k

param h {k in 1..T}; # holding cost for inventory from k-1 to k

param d {k in 1..T}; # demand in period k

# Dynamic Programming Computation:

param MaxQuantity := sum{k in 1..T} d[k];

set allowableDecisions {k in 1..T, I in 0..MaxQuantity by 10} :=
 {x in 0..MaxQuantity by 10: d[k] <= I + x <= MaxQuantity};
 # quantity to produce at stage k, state I (inventory level)

param f {k in 1..T+1, I in 0..MaxQuantity by 10} :=
 if k=T+1 then 0 # no more demand to consider
 else
   min{x in allowableDecisions[k, I]} (c[k]*x + F[k]*(if x <>0 then 1

else 0) + h[k]*(I + x - d[k]) + f[k+1, I + x - d[k]]);
 # min cost in periods k through T+1

param DPvalue := f[1, 0];
 # compute the optimal value
```

```
set opt {k in 1..T, I in 0..MaxQuantity by 10} :=
 {x in allowableDecisions[k, I]: f[k, I] = c[k]*x + F[k]*(if x <> 0

then 1 else 0) + h[k]*(I + x - d[k]) + f[k+1, I + x - d[k]]};
 # optimal decisions
```