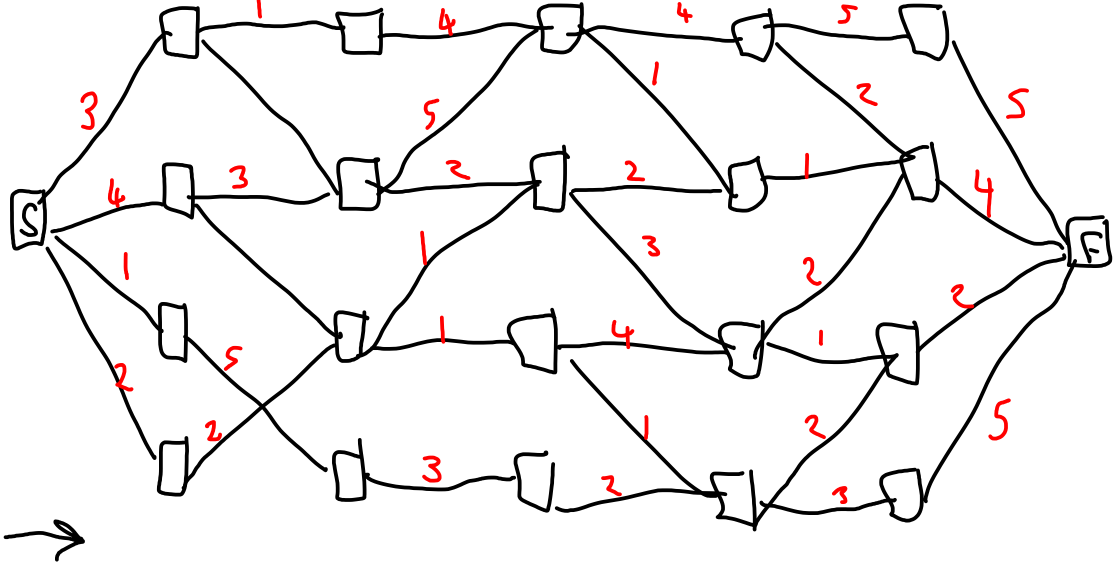
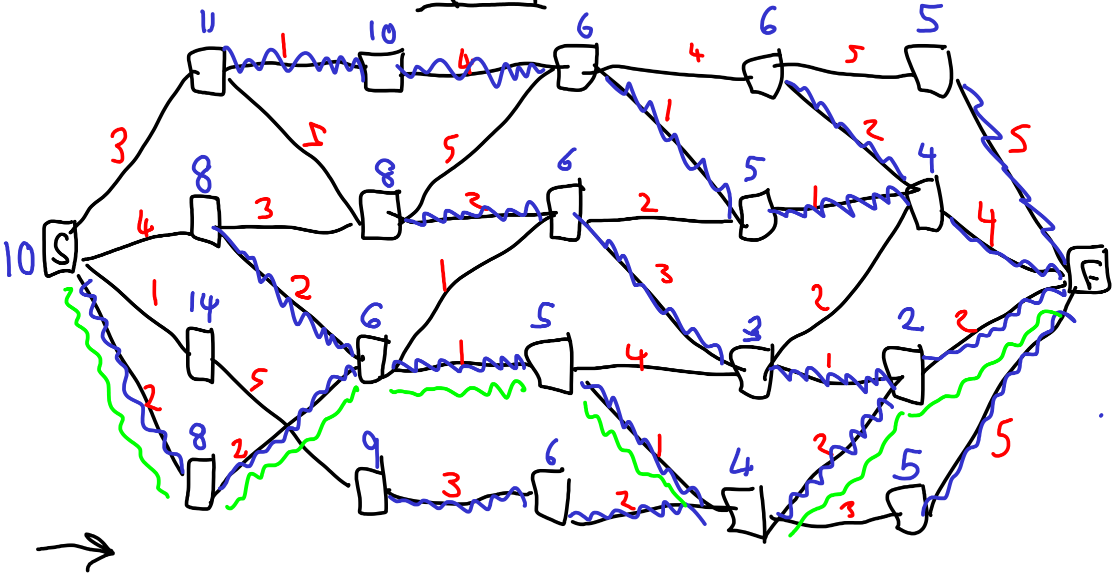
8/25/14

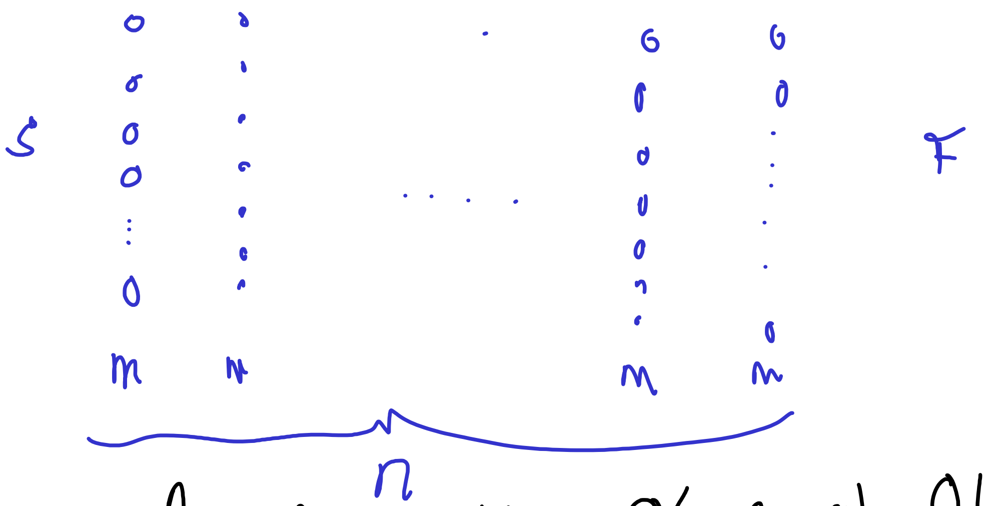
Dynamic Programming

Algorithm methodology


S
F
3
4
1
2
1
3
4
5
2
5
1
2
1
2
4
1
3
1
4
2
2
3
5
2
1
2
4
2
1
2
3
5

8/27/14





\# paths $= m^n$

Running time: $O(mn \times m) = O(m^2 n)$.

Production Scheduling problem

Factory makes a single item e.g. washing machines.

Goal: minimise total production cost

d_j = demand in period j

$c_j(x)$ = cost of making x machines in period j

Can keep up to H in stock at end of period

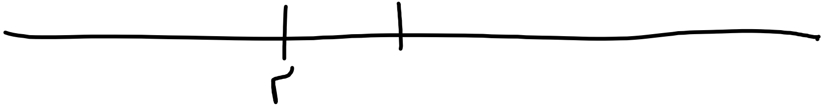
View problem as making a sequence of decisions: $x_1, x_2, \ldots x_n$:

x_j = # machines to make in period j

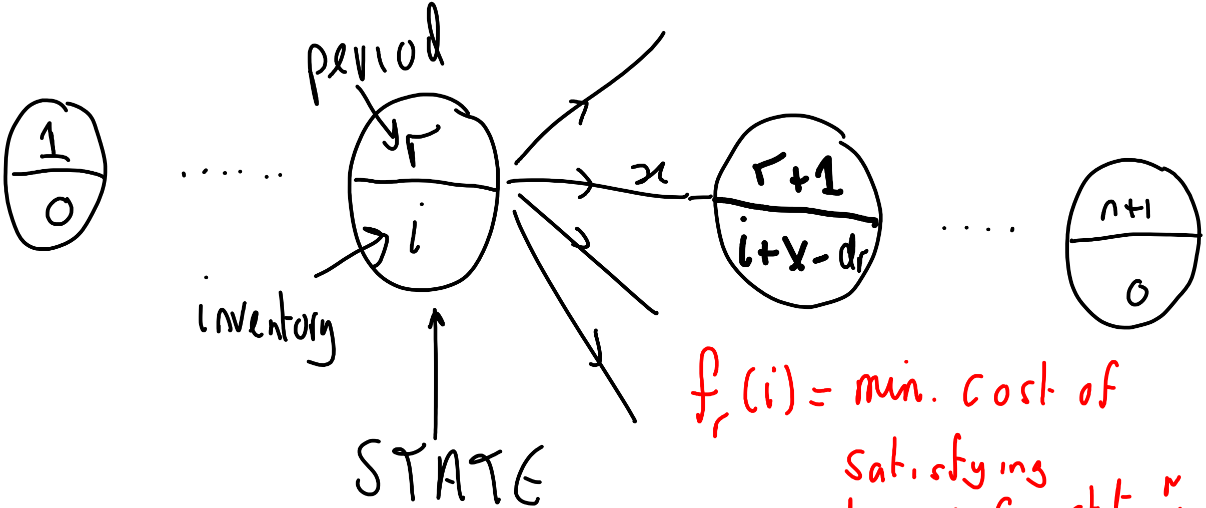
Optimal decion depends on current "state"

What does X_n depend on

Depends on current inventory



Choice of x_r depends on inventory at start of period r.



$f_r(i)$ = min. cost of satisfying demand from state $\genfrac{}{}{0pt}{}{r}{i}$

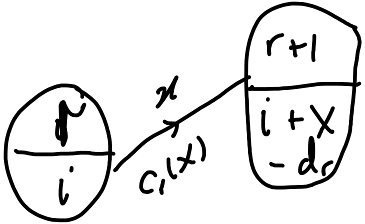
$$f_{n+1}(i) = 0$$

$$f_r(i) = \min_{x} \left(c_r(x) + f_{r+1}(i+x-d_r) \right)$$

$x \geq 0$

$i + x \geq d_r$

$i + x - d_r \leq H$



8/29/14

Example of production problem.

$H = 3$, $n = 5$, $c(x) = 18x - x^2$.

$d_i = 4$, $\forall i$

$$f_r(i) = \min_x \left[c(x) + f_{r+1}(i + x - d_r) \right]$$

t	f_1	x_1	f_2	x_2	f_3	x_3	f_4	x_4	f_5	x_5
0							112 ← 110 ← 164 ← *94 ←	4 5 6 7	56	4
1							101 ← 101 ← 97 ← *89 ←	3 4 5 6	45	3
2					126 ← 134 ← 131 ← 125 ←	2 3 4 5	↗75 Incorrect		32	2
3							↓60		17	1

(i) Add a holding cost: $h(i, x)$

$$f_r(i) = \min_x \{ c(x) + h(i,x) + f_{r+1}(i+x-d_r) \}$$

(ii) Suppose unmet demand is π per item per period — forget holding cost.

$$f_r(i) = \min_x \{ c(x) + \pi(d_r - (i+x))^+ + f_{r+1}(i+x-d_r) \}$$

i can be negative

9\3\14

Example of production problem.

$H = 3$, $n = 5$, $c(x) = 18x - x^2$.

$d_i = 4$, $\forall i$

$$f_r(i) = \min_x \left[c(x) + f_{r+1}(i + x - d_r) \right]$$

i	f_1	x_1	f_2	x_2	f_3	x_3	f_4	x_4	f_5	x_5
0	244	4	188	7	150	4	112 ← 110 ← 164 ← *94 ←	4 5 6 7	56	4.
1	233	3	183	6	139	3	101 ← 101 ← 97 ← *89 ←	3 4 5 6	45	3
2	220	2	176	5	126	2	94	5	32	2
3	205	1	167	1	111	1	89	1	17	1

Variations

① Add a smoothing cost $\sigma(x_1, x_2)$ for making x_1, x_2 in succesive periods

e.g. $\sigma(x_1, x_2) = k(x_1 - x_2)^2$

$$f_r(i, y) = \min_x \left[c(x) + \sigma(y, x) + f_{r+1}(i + x - d_r, x) \right]$$

prod. last period

iii) Machine replacement.

Suppose cost of producing x depends on the age of the machine: $C(X,t)$ for a machine age t.

Cost of new machine is A.

Must acquire new machine once machine reaches T, otherwise it is optional.

$$f_r(i,t) = \min \begin{cases} \text{keep} & \min\limits_{x}\left(C(x,t) + f_{r+1}(i+x-d_r, t+1) \right) \\ \text{replace} & A + \min\limits_{x}\left(C(x,0) + f_{r+1}(i+x-d_r, 1) \right) \end{cases}$$

age of machine

$f_r(i, T) =$

Knapsack problem

Scout X is going to camp.

Has a knapsack.

X can carry at most weight W

n possible items to pack. Each item of type j

has weight w_j and value c_j.

Problem: choose items to maximize value

Integer program: Maximize $c_1 x_1 + c_2 x_2 + \cdots + c_n x_n$

x_i := # items of type i that X takes with them.

$$w_1 x_1 + w_2 x_2 + \cdots + w_n x_n \leq W$$

$$x_i \in \{0, 1, 2, \ldots\}$$

Dynamic Programming: $\mathrm{Opt}[1,2,3,\ldots,n; W] =$

sequence of decisions $x_1, x_2, \ldots$

$$\max_{x_1} \left[c_1 x_1 + \mathrm{Opt}[2,3,\ldots,n; W - w_1 x_1 \right.$$

$f_r(w)$ = max. obtainable using items of type $1, 2, \dots, r$ and knapsack of size w

$$= \max_{x_r} \left[c_r x_r + f_{r-1}(w - w_r x_r) \right]$$

9\05\14

Knapsack Problem:

maximise $c_1x_1 + \cdots + c_nx_n$

s.t. $w_1x_1 + \cdots + w_nx_n \le W$

$x_1, \ldots, x_n \ge 0$ & integer

$$f_r(\omega) = \max.\ c_1x_1 + \cdots + c_rx_r$$

$$w_1x_1 + \cdots + w_rx_r \le \omega$$

$$x_1, \ldots, x_r \ge 0 \text{ \& integer}$$

$$f_r(w) = \max_{0 \le x_r \le \left\lfloor \frac{w}{w_r} \right\rfloor} \left[c_r x_r + f_{r-1}(w - w_r x_r) \right]$$

$$f_1(w) = c_1 \left\lfloor \frac{w}{w_1} \right\rfloor$$

Execution time =

$$O(nW^2)$$

or

$$f_0(w) = 0$$