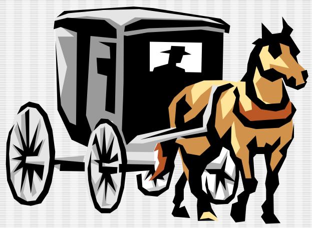
Dynamic Programming

- A general approach to problem-solving
- In most cases: work backwards from the end
- Particular equations must be tailored to each situation
- To develop insight, expose to wide variety of DP problems

Characteristics of DP Problems

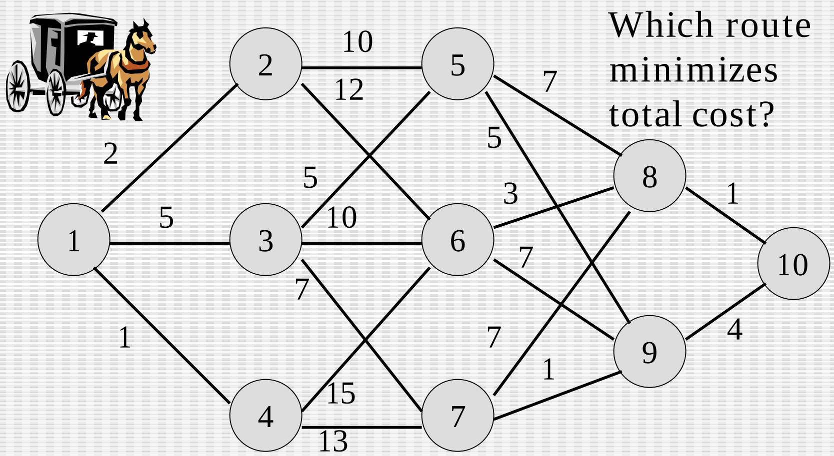
- Stages, decision at each stage
- Each stage has assoc states
- Decision describes transition to next stage
- Given current stage, subsequent decisions must not depend on previously chosen decisions or states.
- Recursion relates cost/reward: $f_t(i) = \min_{j} \{c_{ij} + f_{t+1}(j)\}$

Example: Stagecoach Problem



A fortune-seeker in Missouri decides to go west to join the 49'er gold rush in California. The journey requires traveling by stagecoach through unsettled country, with serious danger of attack by marauders. The starting point and destination are fixed, all possible routes with life-insurance costs are shown on next page.

Stagecoach Problem cont …

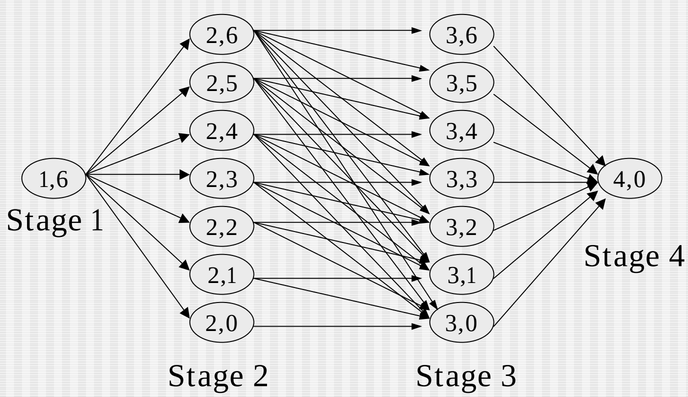


Which route minimizes total cost?

Optimized Investments

- $6000 to invest
- THREE investments – in thousands of $$$
- NPV of returns on each investment, {d1,d2,d3}, is as follows:
  - $r1(d1) = 7d1 + 2$
  - $r2(d2) = 3d2 + 7$
  - $r3(d3) = 4d3 + 5$
  - $r1(0) = r2(0) = r3(0) = 0$
- What are the *stages*, *decision*, *states*? Solve.

Investment Network



Resource Allocation Problem

- w units of resource available
- T activities to which resource can be alloc
- x_t = implementation level of activity t
- $g_t(x_t)$ = resource used by activity t
- $r_t(x_t)$ = benefit for activity t
- Problem:

$$\max \sum_{t=1}^{T} r_t(x_t) \quad \text{s.t.} \quad \sum_{t=1}^{T} g_t(x_t) \le w$$

Resource Allocation Solution

- $f_t^*(d)$ = max benefit from t,t+1,…,T if d units allocated to these activities
- Recursion:

$$f_{T+1}^*(d) = 0 \quad \text{for all } d$$

$$f_t^*(d) = \max_{x_t} \left\{ r_t(x_t) + f_{t+1}^*(d - g_t(x_t)) \right\}$$

Example: Resource Allocation

ASSUMPTIONS

- Amt allocated may be any non-neg no.
- Benefit proportional to amount assigned
- Total benefit is sum of benefits

Work Crew Assignment

Tasks

Workers Assigned	A	B	C	D
0	0	0	0	0
1	5	3	1	5
2	10	7	6	9
3	13	12	13	13
4	14	16	15	17
5	12	16	16	20
6	10	16	17	22

Intro to the Knapsack Problem

- x_i = indicator for item i
- c_i = benefit if i is chosen
- a_i = amt of resource i uses

$$\max z = c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

$$\text{s.t.} \quad a_1x_1 + a_2x_2 + \cdots + a_nx_n \le b$$

$$x_i \in \{0,1\}, i = 1,2,\ldots,n$$

Knapsack Example

- 10-pound knapsack
- Item / Weight / Benefit

Item	Weight	Benefit
1	4 lb	11
2	3 lb	7
3	5 lb	12

- How should the knapsack be filled to maximize benefit?

Knapsack: Alt Recursion

- Before:
  - Stage → item type
  - State → capacity remaining
  - Decision → number of type to include
  - $f^*_t(s) = \max_{d_t} \{r_t(d_t) + f^*_{t+1}(s - g_t(d_t))\}$
- Alternative:
  - Stage/State → capacity of knapsack
  - Decision → which item type to include
  - $g^*(w) = \max_j \{b_j + g(w - w_j) : w_j \leq w, j \in \{1,2,3\}\}$

Knapsack: Turnpike Thm

- Order items according to benefit/unit wt

$$\frac{c_1}{w_1} \geq \frac{c_2}{w_2} \geq \text{☹} \geq \frac{c_n}{w_n}$$

- Use at least one of item 1 if $w \geq w^*$

$$w^* = \frac{c_1 w_1}{c_1 - w_1\left(\frac{c_2}{w_2}\right)}$$

- Used as shortcut for large capacity knapsack

Equipment Replacement E.g.

- Purchasing cost = $1000
- Time horizon = 5 years

Time,t	Maint,m_t	Salvage,s_t
1	60	800
2	80	600
3	120	500

Equipment Replacement E.g.

- Goal: Minimize net costs
- Recursion:
  - c_{tx} = net cost if new machine purchased at time t and operating until time x
  - $g^*(t) = \min_x \{c_{tx} + g^*(x)\}$
  - $t = 0,1,2,3,4$
  - $t+1 \leq x \leq t+3,\ x \leq 5$
- Alternative recursion:
  - Stage → time, t, State → age of machine
  - $f^*_t(x)$ = min cost from t to time 5 given that at time t, machine is x-years-old

Example: Inventory Problem

- Demand = {1,3,2,4}
- Setup cost, K = $3
- Unit cost, c = $1/unit
- Holding cost, h = $0.50/unit
- Max of 5 units can be produced/month
- Max inventory at month's end is 4 units
- Want to MINIMIZE total cost. Solve.

Inventory Example: Network