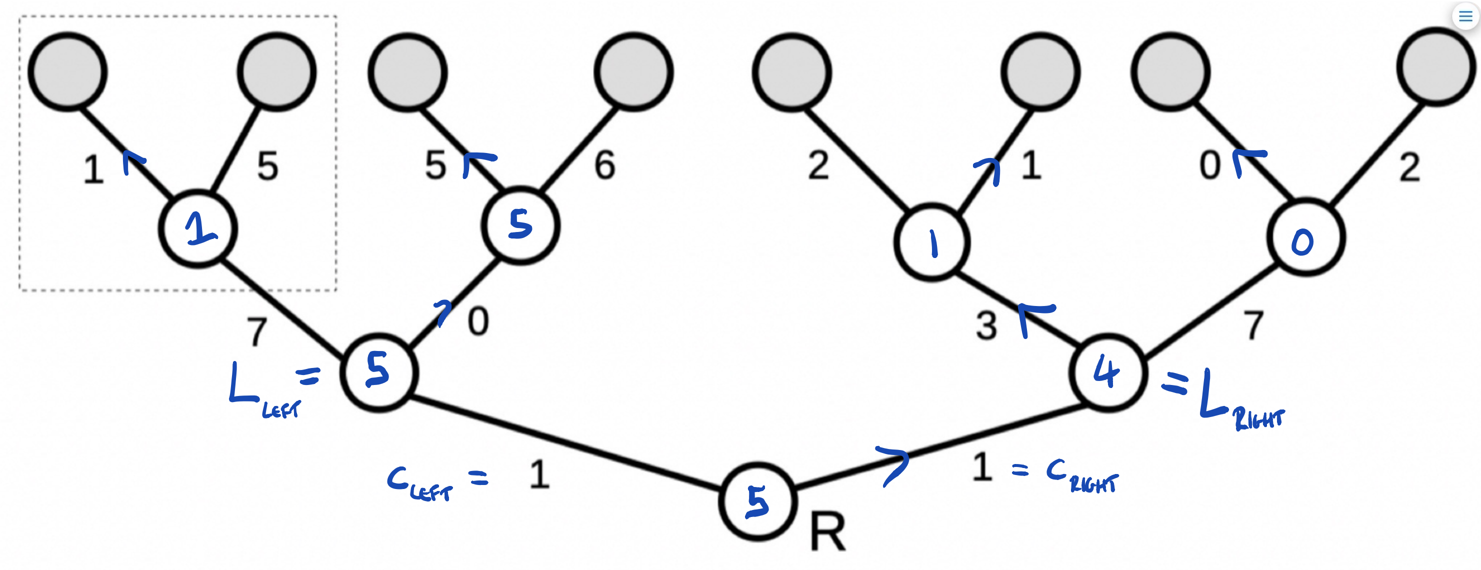
1. DYNAMIC PROGAMMING.

[OPTIMIZING OVER TIME]

Find the Shortest Path from the Root

^ lowest cost



- Solved back-to-front
- Problem repeats itself i.e. solve

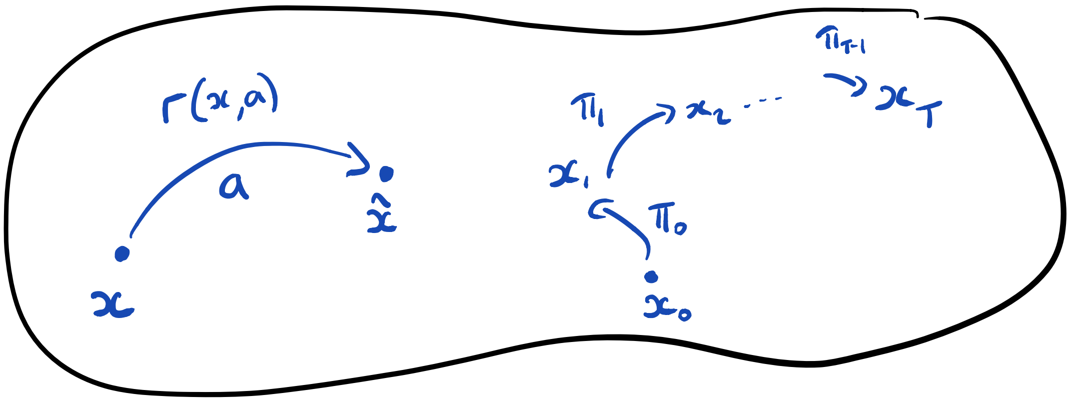
$$L = \min_{a \in \{\text{LEFT}, \text{RIGHT}\}} \{ c_a + L_a \}$$

← [Bellman eqn]

[General solution to shortest paths is Bellman-Ford]

ABSTRACT DEFINITION:

STATE SPACE $\mathcal{X}$



STATE $x \in \mathcal{X}$

ACTION $a \in \mathcal{A}$

REWARD $r(x,a)$

NEXT STATE

$$\hat{x} = f(x,a)$$

POLICY $\pi(x)$ OR π_t .

OBJECTIVE: [MAXIMIZE SUM OF REWARDS]

$$V_T(x_0) := \max \underbrace{\sum_{t=0}^{T-1} r(x_t, \pi_t) + r(x_T)}_{R_T(x,\pi)} \quad \text{OVER } \pi_0, \ldots, \pi_{T-1} \in \mathcal{A}.$$

VALUE FUNCTION

The Bellman Equation

Theorem: $$V_t(x) = \max_{a \in A} \left\{ r(x,a) + V_{t-1}(\hat{x}) \right\}, \qquad V_0(x) = r(x)$$

Proof:

$$V_t(x) = \max_{\underbrace{a_0 \in A, \dots, a_{t-1} \in A}} r(x, a_0) + r(x_1, a_1) + \dots + r(x_{t-1}, a_{t-1}) + r(x_t)$$

$$\max_{a_0 \in A} \max_{a_1, \dots, a_{t-1} \in A}$$

$$= \max_{a_0 \in A} r(x, a_a) + \max_{a_1, \dots, a_{t-1} \in A} \underbrace{r(x_1, a_1) + \dots + r(x_{t-1}, a_{t-1}) + r(x_t)}_{V_{t-1}(\hat{x})}$$

$$= \max_{a_0 \in A} \left\{ r(x, a_0) + V_{t-1}(\hat{x}) \right\} \qquad \square$$

SOME CODE:

```
def DP(time, state, f, r, A):
  '''
  Solves a dynamic program
  '''
  if time > 0 :
    Q = [ r[state][action] + DP(time-1, f[state][action]) for
   action in A ]
    V = max(Q)
  else :
    Q = r[state]
    V = max(Q)
  return V
```

THE PRINCIPLE OF OPTIMALITY:

(aka. WHEN SHOULD DYNAMIC PROGRAMMING WORK!)

"Principle of Optimality: An optimal policy has the property that whatever the initial state and initial decisions are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decisions." – Richard Bellman [2]

EXAMPLE (SHORTEST & LONGEST PATHS)

[SHORTEST PATH GOES THROUGH B FROM WHICH MUST TAKE SHORTEST PATH TO D]

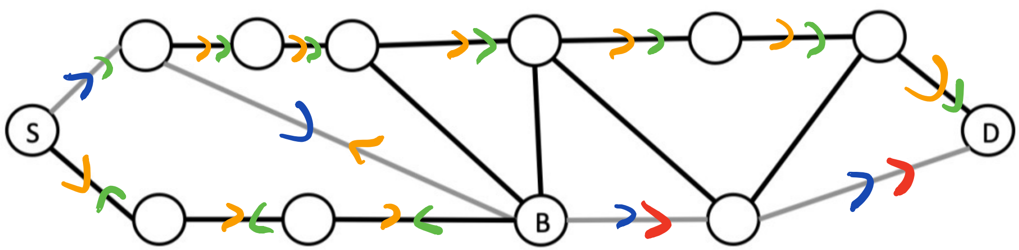
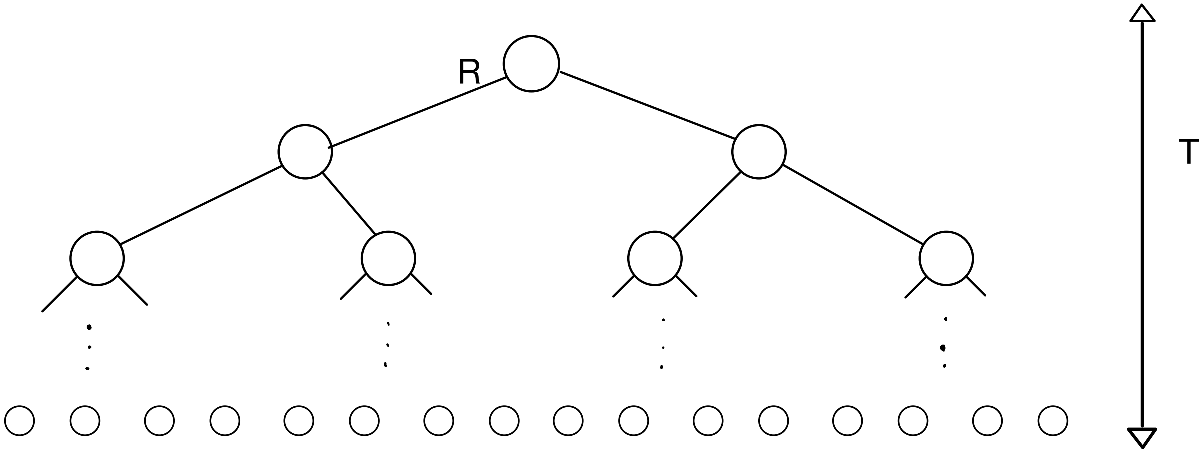


Figure 1.2: Shortest path from S to D.

THE CURSE OF DIMENSIONALITY:

TREE EXAMPLE (AGAIN):

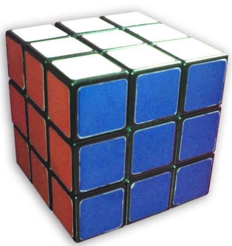


$2^{T+1} - 1$

STATES

TOO BIG!!

RUBIK'S CUBE:



- SOURCE: WIKIPEDIA!

4.3×10^{19}

A SHORTEST PATH PROBLEM

BUT

OPTIMAL SOLUTION UNKNOWN...

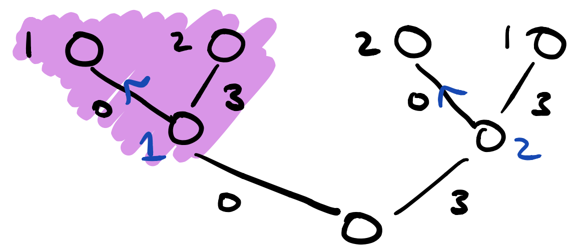
[i.e. WE KNOW HOW TO FIND THE OPTIMAL SOLUTION BUT COMPUTERS AREN'T FAST ENOUGH]

OTHER OBSERVATIONS

MINIMIZE: $L_T(x_0) \doteq \text{MIN} \sum_{t=0}^{T-1} c(x_t, a_t) + c(x_T)$ OVER $a_1, \dots, a_{T-1} \in A$

BELLMAN EQN: $L_t(x_t) = \min_{a \in A} \{ c(x_t, a) + L_{t-1}(\hat{x}) \}$

MEMOIZE: SOME TIMES PROBLEM REPEATS e.g.



SO STORE VALUES & ACTIONS

HELPS OVER COME "CURSE OF DIMENSIONALITY".

GENERALIZE: D.P. STILL WORKS FOR

$$\text{MAX} \sum_{t=0}^{T-1} r_t(x_t, a_t, x_{t+1}) + r_T(x_T)$$

s.t. $a_0 \in A_0, \dots, a_{T-1} \in A_{T-1}$

EXAMPLE: AN INVESTOR HAS A FUND:

- IT HAS x POUNDS AT TIME ZERO
- MONEY CAN'T BE WITHDRAWN
- IT PAYS INTEREST $r \times 100\%$ PER YEAR FOR T YEARS
- THE INVESTOR CONSUMES PROPORTION a_t OF INTEREST & REINVESTS THE REST

WHAT SHOULD YOU DO TO MAXIMIZE CONSUMPTION OVER T YEARS?

ANSWER: LET x_t = TIME t FUND VALUE

WANT TO SOLVE

[BACKWARDS IN TIME]

$$V_T(x) = \max \sum_{t=0}^{T-1} r x_t \cdot a_t$$

s.t. $$x_{t+1} = x_t + \underbrace{r\, x_t (1 - a_t)}_{\text{INTEREST NOT CONSUMED.}}$$

OVER $$0 \leq a_t \leq 1$$

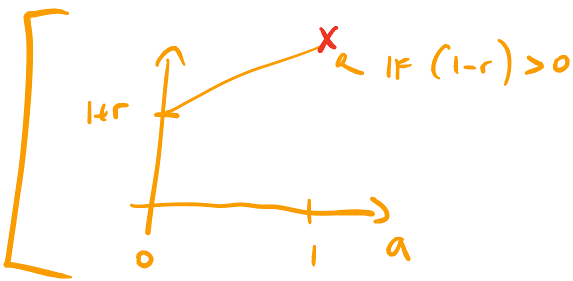
- IF IT'S THE LAST YEAR, CONSUME EVERYTHING!

$$a_{T-1} = 1 \quad \therefore \quad V_1(x) = rx$$

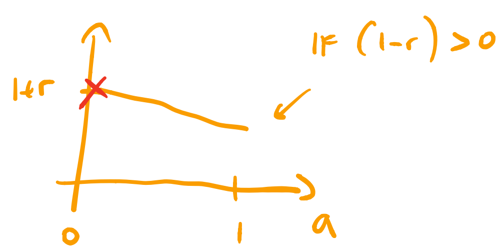
- IF 2 YEARS LEFT,

$$V_2(x) = \max_{0 \le a \le 1} \left\{ rxa + \underbrace{V_1(x + rx(1-a))}_{rx + r^2 x(1-a)} \right\}$$

$$= rx \max_{0 \le a \le 1} \left\{ a(1-r) + (1+r) \right\} = rx \cdot \underbrace{2 \vee (1+r)}_{\rho_2}$$



OR



CALLED BANG-BANG CONTROL.

- LETS SOLVE FOR $t+1$ YEARS LEFT.

A GUESS

$$V_t(x) = r x \rho_t \qquad \text{FOR SOME } \rho_t$$

PROOF BY INDUCTION:

$$V_{t+1}(x) = \max_{0 \le a \le 1} \Big\{ r x a + \underbrace{V_t\,(x + r x(1-a))}_{\rho\,(x + r x(1-a))} \Big\}$$

$$= r x \max_{0 \le a \le 1} \big\{ a(1 - r\rho) + \rho(1+r) \big\}$$

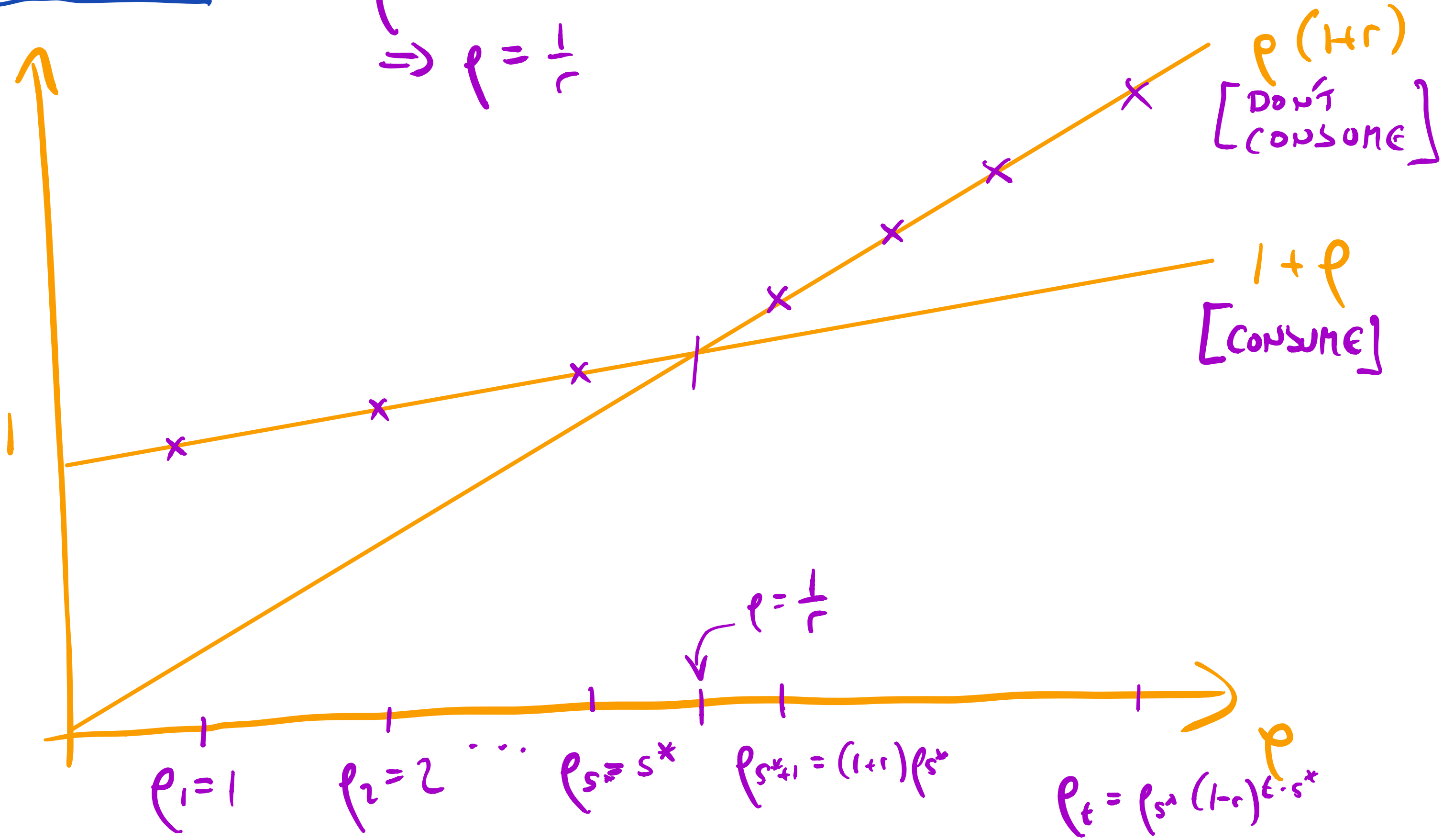
$$= r x \cdot \{\underbrace{\rho(1+r)}_{a=0}\} \vee \underbrace{(1+\rho)}_{a=1}$$

SO

$$\rho_{t+1} = r x \cdot \quad \rho_t(1+r) \vee (1+\rho)$$

A PICTURE:

$$\rho(1+r) = 1+\rho \implies \rho = \frac{1}{r}$$



OPTIMAL: TO CONSUME FOR LAST $\lfloor \frac{1}{r} \rfloor$ YEARS

OTHERWISE REINVEST ALL INTEREST.

SUMMARY:

- DP : $V_t(x_0) \doteq \max_{a_0,\dots,a_{t-1} \in A} r(x_0, a_0) + \dots + r(x_{t-1}, a_{t-1}) + r(x_t)$

- BELLMAN EQN : $V_t(x) = \max_{a \in A} \left\{ r(x,a) + V_{t-1}(\hat{x}) \right\}$, $V_0(x) = r(x)$

- BELLMAN EQN – CAN BE SOLVED INDUCTIVELY $(t = 0, 1, 2, \dots)$
  – COMPLEXITY SCALES BADLY (BUT LOTS OF TRICKS EXIST eg memoizing, function approximation.)

- PRINCIPLE OF OPTIMALITY – TO SOLVE AN OPTIMIZATION, ALWAYS TAKE OPTIMAL DESCIONS FROM EACH STATE ONWARDS