Finite horizon Problems :-

By a controlled Markov chain we mean a sequence of Random Variables on a common probability space $\{X_n\}$ taking values in the same set S and whose dynamics depends on a control sequence $\{Z_n\}$ s.t. $\forall\, Z_n \in A \;\; \forall\, n$ for same set A

★ **Controlled Markov property :-**

$$P(X_{n+1} = j \mid X_n = i, Z_n = z, X_{n-1} = i_{n-1}, Z_{n-1} = z_{n-1}, \ldots, X_0 = i_0, Z_0 = z_0)$$

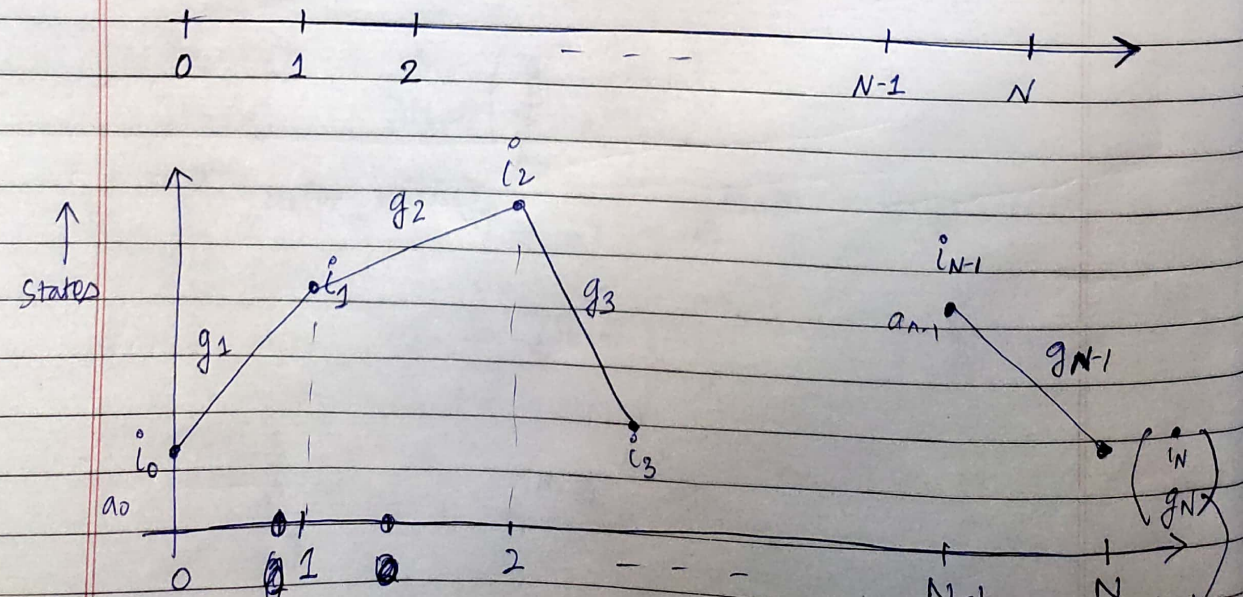
$$= P(X_{n+1} = j \mid X_n = i, Z_n = z) \triangleq p(i, z, j)$$

★ **Markov Decision Process :- (MDP) :-**

A Markov Decision Process (MDP) is a controlled Markov process with an additional cost structure.

- A cost $g(i, z, j)$ is incurred when $X_n = i,\ Z_n = z,\ X_{n-1} = j$

Let N be the horizon (length), $N < \infty$



Page No.
Date / /20

* Long term cost objective :-

$$J_{\{Z_0, Z_1, \ldots Z_{n-1}\}}(i_0) = E\left[g_N(X_N) + \sum_{k=0}^{N-1} g_k(X_k, Z_k, X_{k+1}) \mid X_0 = i_0 \right]$$

Here, $E[\cdot]$ is the expectation over the joint distribution of the next states, $X_1, X_2, \ldots, X_N$ given that $X_0 = i_0$.

* Policy :- Decision rule specified is a set of functions that suggests the action to chose at a given instant.

* Let $\pi = \{\mu_0, \mu_1, \ldots \mu_{N-1}\}$ where $\mu_k : S \to A$

$\forall\ k = 0, 1, \ldots N-1$

* We say that π is an admissible policy if
$\mu_k : S \to A$
$\mu_k(i) \to A(i)$ [set of feasible actions in state i]
$A(i) \subset A \quad \forall\ i \in S$ & $\bigcup_{i \in S} A(i) = A$

* Deterministic Policy.
* Stochastic Policies.

* Optimal Policy :- An optimal policy π^* is one for which

$$J_{\pi^*}(i_0) = \min_{\pi \in \Pi} J_\pi(i_0) \quad \forall\ i_0 \in S.$$

where Π = set of all admissible policies.

Here,

$$J_\pi(i_0) = E\left[g(X_N) + \sum_{k=0}^{N-1} g_k(X_k, \mu_k(X_k), X_{k+1}) \mid X_0 = i_0 \right]$$

Here $\pi = \{\mu_0, \mu_1 \ldots \mu_{N-1}\}$

We also refer to the optimal cost as

$J^*(i_0) = J_{\pi^*}(i_0)$.

$J^*(i_0)$: optimal cost. $J_{\pi^*}(i_0)$: Optimal policy.

$$\pi^* = \underset{\pi \in \Pi}{\operatorname{argmin}}\ J_\pi(i_0)$$

Let $\pi^* = \{\mu_0^*, \mu_1^*, \dots, \mu_{N-1}^*\}$ be the optimal policy. Assume that under π^*, a state x_i occurs at time i with positive probability.
Consider the following subproblem:

$$\min_{\pi^i = \{\mu_i, \mu_{i+1}, \dots, \mu_{N-1}\}} E\left[g_N(x_N) + \sum_{k=i}^{N-1} g_k(x_k, \mu_k(x_k), x_{k+1}) \,\middle|\, x_i = x_i \right]$$

Then the truncated policy:-

$$\pi^{*i} = \{\mu_i^*, \mu_{i+1}^*, \dots, \mu_{N-1}^*\}$$

is optimal for the subproblem.

---

* Principle of Optimality:-

This is called Principle of optimality suggests a dynamic programming algo.

Proposition:- For every initial state x_0, the optimal cost $J^*(x_0) = J_0(x_0)$ that is given by the last step of the

following algo. that goes backward in time.

$$J_N(x_N) = g_N(x_N). \quad \forall x_N \in S \longrightarrow (1)$$

$$J_k(x_k) = \min_{a_k \in A(x_k)} E_{x_{k+1}}\left[g_k(x_k, a_k, x_{k+1}) + J_{k+1}(x_{k+1}) \right]$$

$$\forall x_k \in S$$

$$\forall k = N-1, N-2 \ldots 0$$

$$\longrightarrow (2)$$

Also, if $\mu_k^* = \mu_k^*(x_k)$ minimizes the RHS of (2), $\forall \mu_k \; \forall k$, the policy $\pi^* = \{\mu_0^*, \mu_1^* \cdots \mu_{N-1}^*\}$ is optimal.

Let's look at (1) & (2) in detail :-

$$J_N(x_N) = g_N(x_N). \quad \forall x_N \in S$$

Randomness

$$J_{N-1}(x_{N-1}) = \min_{a_{N-1} \in A(x_{N-1})} E_{x_N}\left[g_{N-1}(x_{N-1}, a_{N-1}, x_N) + J_N(x_N) \right]$$

---

* Proof :- Why does this DP algorithm converges.

Let for any admissible policy $\pi = \{\mu_0, \mu_1, \cdots, \mu_{N-1}\}$

$\pi^k = \{\mu_k, \mu_{k+1}, \cdots \mu_{N-1}\}$, $k = 0, 1, \cdots N-1$ denote the k^{th} tail policy.

let

$$J_k^*(x_k) = \min_{\pi^k} E_{x_{k+1}, \cdots x_N}\left[g_N(x_N) + \sum_{i=k}^{N-1} g_i(x_i, \mu_i(x_i), x_{i+1}) \,\middle|\, x_k = x_k \right]$$

$$, k = 0, 1, \cdots N-1.$$

Also, let $J_N^*(x_N) = g_N(x_N) = J_N(x_N) \quad \forall x_N$.

We show by induction that $J_k^*(x_k) \; J_k(x_k) \quad \forall k, \forall x_k$.

Assume that for some k and all x_{k+1}, $J_{k+1}^*(x_{k+1}) = J_{k+1}(x_{k+1})$.

Note: $\pi^k = \{\mu_k, \pi^{k+1}\}$, Thus $\forall x_k$

$$J_k^*(x_k) = \min_{\{\mu_k, \pi^{k+1}\}} E_{x_{k+1}, \ldots, x_N}\Big[g_N(x_N) + g_k(x_k, \mu_k(x_k), x_{k+1}) + \sum_{i=k+1}^{N-1} g_i(x_i, \mu_i(x_i), x_{i+1}) \;\Big|\; x_k = x_k\Big]$$

$$= \min_{\{\mu_k, \pi^{k+1}\}} E_{x_{k+1}}\Big[g_k(x_k, \mu_k(x_k), x_{k+1}) + E_{x_{k+2}, \ldots, x_N}\Big[g_N(x_N) + \sum_{i=k+1}^{N-1} g_i(x_i, \mu_i(x_i), x_{i+1}) \;\Big|\; x_{k+1}\Big] \;\Big|\; x_k = x_k\Big]$$

$$= \min_{\mu_k} E_{x_{k+1}}\Big[g_k(x_k, \mu_k(x_k), x_{k+1}) + \min_{\pi^{k+1}} E_{x_{k+2}, \ldots, x_N}\Big[g_N(x_N) + \sum_{i=k+1}^{N-1} g_i(x_i, \mu_i(x_i), x_{i+1}) \;\Big|\; x_{k+1}\Big], \;\Big|\; x_k = x_k\Big]$$

$$J_{k+1}^*(x_{k+1})$$

$$= \min_{\mu_k} E_{x_{k+1}} [g_k(x_k, \mu_k(x_k), x_{k+1}), J^*_{k+1}(x_{k+1}) \mid x_k = x_k].$$

From induction hypothesis $J^*_{k+1}(x_{k+1}) = J_{k+1}(x_{k+1})$.

$$J^*_k(x_k) = \min_{\mu_k} E_{x_{k+1}} [g_k(x_k, \mu(x_k), x_{k+1}) + J_{k+1}(x_{k+1}) \mid x_k = x_k].$$

$$= \min_{q_k \in A(x_k)} E_{x_{k+1}} [g_k(x_k, q_k, x_{k+1}) + J_{k+1}(x_{k+1}) \mid x_k = x_k)$$

$$= J_k(x_k).$$

$\Rightarrow J^*_j(x_j) = J_j(x_j) \quad \forall \quad j = 0, 1, \ldots N \quad \forall x_j$

Dyn. Prog & Opt. Control

Finite Horizon Problems :-

In general,

S_k = state space at time k.
A_k = Action space ___ ___

$A_k(x_k)$ = set of feasible actions at time k in state x_k.

S - State space.
A - Action Space.
x_k = state at time k.
a_k = action at ___ ___

$a_k \in \underbrace{A(x_k)}_{\text{Set of feasible actions in } x_k} \subset A$

$g_k(x_k, \mu_k, x_{k+1})$ - simple state cost at $k = 0, 1, \ldots N-1$

$g_N(x_N)$ - terminal cost.

* Dynamic Programming Algorithm :-

$$J_N(x_N) = g_N(x_N) \quad \forall\, x_N \in S.$$

$$J_k(x_k) = \min_{a_k \in A(x_k)} E_{x_{k+1}} \left[g_k(x_k, a_k, x_{k+1}) + J_{k+1}(x_{k+1}) \right]$$

$$\forall\, k = N-1, \ldots 0$$
$$\forall\, x_k \in S.$$

$$= \min_{a_k \in A(x_k)} \sum_{j \in S} p(x_k, a_k, j) \left(g_k(x_k, a_k, j) + J_{k+1}(j) \right)$$

* Inf

*

Page No.
Date / /20

Analogy for infinite horizon.

$$J(x) = \min_{a \in A(x)} \sum_{0} p(x,a,j)\,[\,g(x,a,j) + \gamma J(j)\,].$$

$\gamma \in (0,1)$

* Infinite horizon problem (discounted cost).

Suppose π is policy.

$$V_\pi(x) = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k g(x_k, \pi(x_k), x_{k+1}) \,\middle|\, x_0 = x\right]$$

(value of stat x under policy π)

Suppose $|g(x_k, \pi(x_k), x_{k+1})| \leq B < \infty$

$$\Rightarrow |V_\pi(x)| \leq \frac{B}{(1-\gamma)} < \infty$$

* Long run avg costs :-

$$J(\pi) = \lim_{N\to\infty} \frac{1}{N} E_\pi\left[\sum_{k=0}^{N-1} g(x_k, \pi(x_k), x_{k+1}) \,\middle|\, x_0 = x\right]$$

Exists if under π the markov chains is ergodic.

$$\boxed{J(\pi) = \lim_{\gamma \to 1} (1-\gamma) V_\pi(x)}$$

★ Example (Finite Horizon) :-

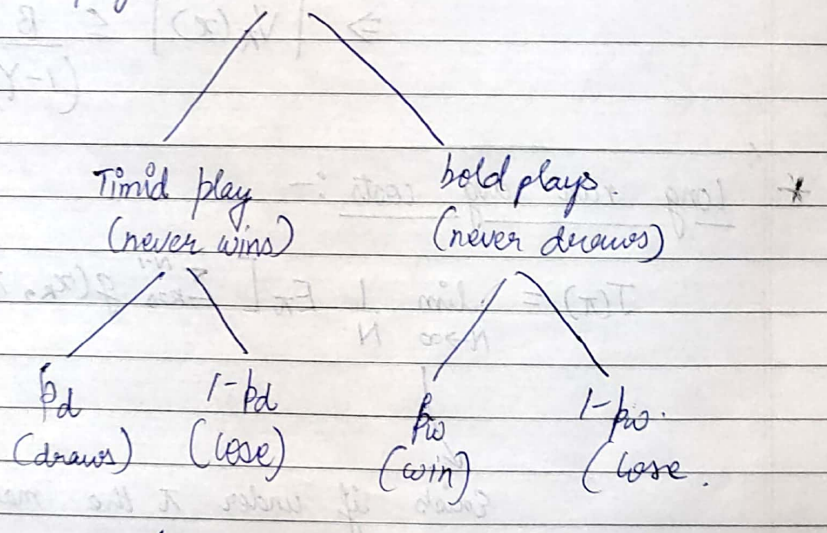
There is a player playing chess with an opponent. Total of N games to be played in match. If scores are tied after N games, then match goes in sudden death mode. (first player after N games to win, wins the match).

Score assignment :- Win — 1 pt.
Loss — 0 pt.
Draw — 1/2 pt.

State = net score = pts of player − pts of opponent.

Note :- Maximization problem.

Action :- A player can select



Assume :- $r_k(x_k, u_k, x_{k+1}) = 0 \quad \forall\ k = 0, 1, \ldots N-1$

$$r_N(x_N) = \begin{cases} 1 & ; x_N > 0 \\ p_w & ; x_N = 0 \\ 0 & ; x_N < 0 \end{cases}$$

Page No.
Date / /20

Dynamic Programming :- eqn :-

$$J_k(x_k) = \max \left[p_d \cdot J_{k+1}(x_k) + (1-p_d) J_{k+1}(x_k - 1), \right.$$
$$\left. p_w J_{k+1}(x_k+1) + (1-p_w) J_{k+1}(x_k - 1) \right]$$

Thus it is optimal to play bold if,

$$p_w J_{k+1}(x_k+1) + (1-p_w) J_{k+1}(x_k - 1).$$
$$\geq p_d J_{k+1}(x_k) + (1-p_d) J_{k+1}(x_k - 1).$$

$$p_w \cdot J_{k+1}(x_k+1) + J_{k+1}(x_k - 1) - p_w \cdot J_{k+1}(x_k - 1)$$
$$\geq p_d \cdot J_{k+1}(x_k) + J_{k+1}(x_k - 1) - p_d \left(J_{k+1}(x_k - 1) \right)$$

$$\Rightarrow \frac{p_w}{p_d} \geq \frac{J_{k+1}(x_k) - J_{k+1}(x_k - 1)}{J_{k+1}(x_k+1) - J_{k+1}(x_k - 1)}$$

Suppose $p_d > p_w$.

Consider $k = N-1$

Consider $k = N-1$

$$J_{N-1}(x_{N-1}) = \max \left[p_d \cdot J_N(x_{N-1}) + (1-p_d) \cdot J_N(x_{N-1} - 1), \right.$$
$$\left. p_w \cdot J_N(x_{N-1}+1) + (1-p_w) J_N(x_{N-1} - 1) \right]$$

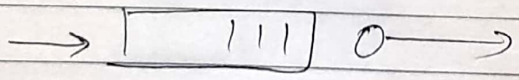
$$J_{N-1}(x_{N-1}) = \begin{cases} 1 & \text{if } x_{N-1} > 1 \text{ - Either is optimal} \\ p_d + (1-p_d)p_w & \text{if } x_{N-1} = 1 \text{ - Play timid} \\ p_w & \text{if } x_{N-1} = 0 \text{ - Play bold} \\ p_w^2 & \text{if } x_{N-1} = -1 \text{ - play bold} \\ 0 & \text{if } x_{N-1} < -1 \end{cases}$$

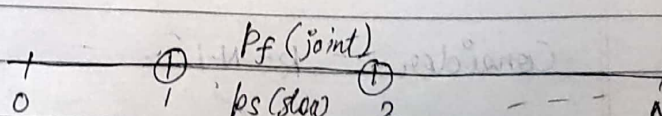
For 2 games remaining :-

$$J_{N-2}(x_{N-2}) = \max \left\{ \begin{array}{l} p_d \cdot J_{N-1}(x_{N-2}) + (1-p_d) \cdot J_{N-1}(x_{N-2}) \\ p_w \cdot J_{N-1}(x_{N-2}+1) + (1-p_w) \cdot J_{N-1}(x_{N-2}) \end{array} \right]$$

$$J_{N-2}(x_{N-2}) = \begin{cases} p_w & \text{if } x_{N-2} = 0 \text{ — play bold.} \\ p_d + (1-p_d) \cdot J_{N-1}(x_{N-2}-1) & \text{if } x_{N-2} > 0 \rightarrow \text{play timid.} \end{cases}$$

Ex.





Buffer size = n

probability that m customers come in at an instant = p_m.

Page No.
Date / /20

Customer served as per FCFS;-

Types of service → fast :- cost C_f — exits w.p. p_f ($C_f > C_s$)
→ slow :- cost C_s — exits w.p. p_s ($p_f > p_s$)

If at end of a period, i customers in system.
holding cost = $r(i)$.

If at the end of the horizon, i customers in system, then cost → $R(i)$.

Assume no. of arrivals in a period independent of no. of arrivals in any other period.

Transition Prob.:-

$$p_{oj}(\mu_f) = p_j = p_{oj}(\mu_s). \quad j = 0, 1, \dots, n-1.$$

$$p_{on}(\mu_f) = \sum_{j=n}^{\infty} p_j = p_{on}(\mu_s).$$

for $i > 0$

$$p_{ij}(\mu_f) = 0 \quad \text{if } j < i-1.$$

$$p_{ij}(\mu_f) = p_f p_0 \quad \text{if } j = i-1$$

(p_f: fast/slow, p_0: no. of arrival of people.)

$$p_{ij}(\mu_f) = P(i-j+1 \text{ arrivals, service completion}) \quad i-1 < j < n-1$$
$$+ P(j-i \text{ arrivals, no service completion})$$
$$= p_f p_{j-i+1} + (1-p_f) p_{j-i}$$

$j = n-1$

$$p_{ij}(\mu_f) = p_f \sum_{m=n-i}^{\infty} p_m + (1-p_f)\, p_{n-1-i}$$

(1)

$j = n$

$$p_{ij}(\mu_f) = (1-p_f) \sum_{m=n-i}^{\infty} p_m$$

DP Algo :-

$$J_N(i) = R(i), \quad i = 0, 1, \ldots n.$$

$$J_k(x_k) = \min \left[r(i) + C_f + \sum_{j=0}^{\infty} p_{ij}(\mu_f)\, J_{k+1}(j), \right.$$
$$\left. r(i) + C_s + \sum_{j=0}^{\infty} p_{ij}(\mu_s)\, J_{k+1}(j) \right]$$

(2)