Dynamic programming and optimal control

- Course program
  - Seven chapters, seven lectures
  - Seven problem solving sessions
- Examination

Schedule

1. The dynamic programming algorithm (MN)
2. Deterministic systems and the shortest path problem (FT)
3. Deterministic continuous-time optimal control
4. Problems with perfect state information
5. Problems with imperfect state information
6. Suboptimal and adaptive control
7. Introduction to infinite horizon problems

Model for general finite horizon optimal control

1. *Discrete time dynamic system*

$$x_{k+1} = f_k(x_k, u_k, w_k), \qquad k = 0, 1, \dots, N-1$$

2. *Cost function*

$$J = E\Big\{g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k)\Big\}$$

Example (Ticket salesman)

- x_k tickets available
- u_k tickets ordered (and delivered) ($u_k \geq 0$)
- w_k demand for tickets

$w_0, \dots, w_{N-1}$ independent random variables.
The state equation

$$x_{k+1} = x_k + u_k - w_k$$

example, cont'd

The cost consists of

1. cost $r(x_k)$ for ticket stock
2. purchase of tickets, cu_k
3. terminal cost $R(x_N)$.

This leads to the over-all cost

$$E\Big\{R(x_N) + \sum_{k=0}^{N-1} \Big(r(x_k) + cu_k\Big)\Big\}$$

Optimization strategy

- Open-loop.
  Decide the control, $u_0, \dots, u_{N-1}$, immediately at time zero.
- Closed-loop.
  Use all information available, delay decision of u_k until time k.

Note: The goal in closed loop becomes to find an optimal rule for selecting u_k regardless of the value of x_k not only a numerical value (*strategy vs. action*).

System description

$$x_{k+1} = f_k(x_k, u_k, w_k), \qquad k = 0, 1, \dots, N-1$$

x_k can be real or discrete as in the ticket example.

Other systems has a more natural representation as states with probabilities for transitions between the states.

$$p_{ij}(u, k) = P\{x_{k+1} = j | x_k = i, u_k = u\}$$

which can be expressed as

$$x_{k+1} = w_k$$

with the probability distribution for w_k given as

$$P\{w_k = j | x_k = i, u_k = u\} = p_{ij}(u, k)$$

Conclusion: A system can be represented as either a difference equation or states with transition probabilities.

Example, a machine

A machine can be in n states, $1 \le x_k \le n$, where a lower state is "better" than a higher state. $x_k = 1 \Rightarrow$ perfect machine. The operating cost is $g(x_k)$ and

$$g(1) \le g(2) \le \dots \le g(n)$$

The transition probability is p_{ij} and $p_{ij} = 0$ if $j < i$.

At each stage in the process, the problem is to decide if to

1. let the machine operate one more period
2. repair the machine and bring it to $x_{k+1} = 1$ at a cost R.

The basic DP problem

1. Given

$$x_{k+1} = f_k(x_k, u_k, w_x), \qquad k = 0, 1, \ldots, N-1$$

   with $x_k \in S_k$, $u_k \in U(x_k) \subset C_k$, and $w_k \in D_k$. The random disturbance is characterized by the probability distribution, $P_k(\cdot|x_k, u_k)$.

2. A class of policies (control laws) is a sequence of functions

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

   where $u_k = \mu_k(x_k)$. If $\mu_k(x_k) \in U(x_k)$ for all $x_k \in S_k$ the policy is called *admissible*.

Given x_0 and $\pi = \{\mu_0, \ldots, \mu_{N-1}\}$ the system equation becomes

$$x_{k+1} = f_k(x_k, \mu_k(x_k), w_k), \qquad k = 0, 1, \ldots, N-1$$

Expected cost

$$J_\pi(x_0) = \underset{\substack{w_k \\ k=0,1,\ldots,N-1}}{E} \left\{ g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k), w_k) \right\}$$

An *optimal policy* π^*

$$J_{\pi^*}(x_0) = \min_{\pi \in \Pi} J_\pi(x_0)$$

Often π^* optimal for all initial conditions, therefore

$$J^*(x_0) = \min_{\pi \in \Pi} J_\pi(x_0)$$

with $J^*(x_0)$ the *optimal cost function* or the *optimal value function*.

The dynamic programming algorithm

Principle of optimality
Let $\pi^* = \{\mu_0^*, \mu_1^*, \ldots, \mu_{N-1}^*\}$ be an optimal policy.
Assume that when using π^*, a given state x_i occurs with a positive probability.
Consider the subproblem, starting in x_i and minimizing the "cost-to-go" from time i to time N

$$E\left\{ g_N(x_N) + \sum_{k=i}^{N-1} g_k(x_k, \mu_k(x_k), w_k) \right\}$$

For this subproblem the truncated policy $\{\mu_i^*, \mu_{i+1}^*, \ldots, \mu_{N-1}^*\}$ is optimal!

the DP algorithm, cont'd

For every initial state x_0, the optimal cost $J^*(x_0)$ of the basic problem is equal to $J_0(x_0)$. $J_0(x_0)$ is given by

1. $J_N(x_N) = g_N(x_N)$.
2. For $k = 0, 1, \ldots, N-1$,

$$J_k(x_k) = \min_{u_k \in U(x_k)} \underset{w_k}{E} \{ g_k(x_k, u_k, w_k) + J_{k+1}(f_k(x_k, u_k, w_k)) \}$$

If $u_k^* = \mu_k^*(x_k)$ minimizes the right side of 2 for each x_k and k, then the policy $\pi^* = \{\mu_0^*, \ldots, \mu_{N-1}^*\}$ is optimal.
Proof by induction.

Example, finite state system

The system description

$$x_{k+1} = w_k$$

and the prob. distribution for w_k

$$P\{w_k = j | x_k = i, u_k = u\} = p_{ij}(u)$$

The DP algorithm becomes

$$J_k(i) = \min_{u \in U(i)} [g(i,u) + E\{J_{k+1}(w_k)\}]$$

In the machine example:

$$J_N(i) = 0$$

$$J_k(i) = \min \left[R + g(i) + J_{k+1}(1), g(i) + \sum_{j=i}^{n} p_{ij} J_{k+1}(j) \right]$$

Some notes on the DP algorithm

- A large number of models give analytical optimal solutions.
- The solutions to simple examples may give a basis for suboptimal control schemes in more complex cases.
- The analytically solvable models provides guidelines for modeling.
- In many practical cases analytical solutions is not possible to find. Numerical solution is approximate and time-consuming. (Ref Chapter 6).

State augmentation

- Time lags

  $$x_{k+1} = f_k(x_k, x_{k-1}, u_k, u_{k-1}, w_k), \qquad k = 1, 2, \ldots, N-1$$

- Correlated disturbances, linear systems representation

  $$w_k = C_k y_{k+1}, \qquad y_{k+1} = A_k y_k + \zeta_k, \qquad k = 0, 1, \ldots, N-1$$

- Forecasts
  Example: The controller receives an accurate prediction that the next disturbance w_k will be selected according to a particular probability distribution (from a given collection of distributions).