6.231: DYNAMIC PROGRAMMING

LECTURE 1

LECTURE OUTLINE

- Problem Formulation
- Examples
- The Basic Problem
- Significance of Feedback

DP AS AN OPTIMIZATION METHODOLOGY

- Generic optimization problem:

$$\min_{u \in U} g(u)$$

where u is the optimization/decision variable, $g(u)$ is the cost function, and U is the constraint set

- Categories of problems:
  - Discrete (U is finite) or continuous
  - Linear (g is linear and U is polyhedral) or nonlinear
  - Stochastic or deterministic: In stochastic problems the cost involves a stochastic parameter w, which is averaged, i.e., it has the form

$$g(u) = E_w\{G(u, w)\}$$

where w is a random parameter.

- DP can deal with complex stochastic problems where information about w becomes available in stages, and the decisions are also made in stages and make use of this information.

BASIC STRUCTURE OF STOCHASTIC DP

- Discrete-time system

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

  - k: Discrete time
  - x_k: State; summarizes past information that is relevant for future optimization
  - u_k: Control; decision to be selected at time k from a given set
  - w_k: Random parameter (also called disturbance or noise depending on the context)
  - N: Horizon or number of times control is applied

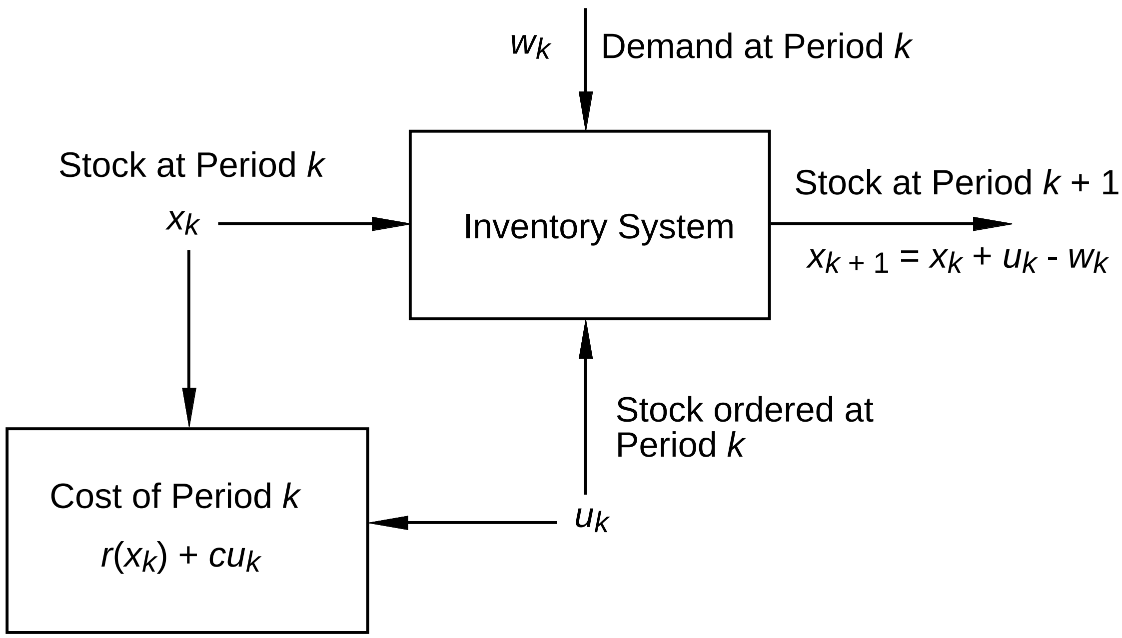
- Cost function that is additive over time

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

- Alternative system description: $P(x_{k+1} \mid x_k, u_k)$

$$x_{k+1} = w_k \quad \text{with } P(w_k \mid x_k, u_k) = P(x_{k+1} \mid x_k, u_k)$$

INVENTORY CONTROL EXAMPLE



- Discrete-time system

$$x_{k+1} = f_k(x_k, u_k, w_k) = x_k + u_k - w_k$$

- Cost function that is additive over time

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

$$= E\left\{\sum_{k=0}^{N-1} \big(cu_k + r(x_k + u_k - w_k)\big)\right\}$$

- Optimization over policies: Rules/functions $u_k = \mu_k(x_k)$ that map states to controls

ADDITIONAL ASSUMPTIONS

- The set of values that the control u_k can take depend at most on x_k and not on prior x or u
- Probability distribution of w_k does not depend on past values $w_{k-1}, \ldots, w_0$, but may depend on x_k and u_k
  - Otherwise past values of w or x would be useful for future optimization
- Sequence of events envisioned in period k:
  - x_k occurs according to

$$x_k = f_{k-1}\big(x_{k-1}, u_{k-1}, w_{k-1}\big)$$

  - u_k is selected with knowledge of x_k, i.e.,

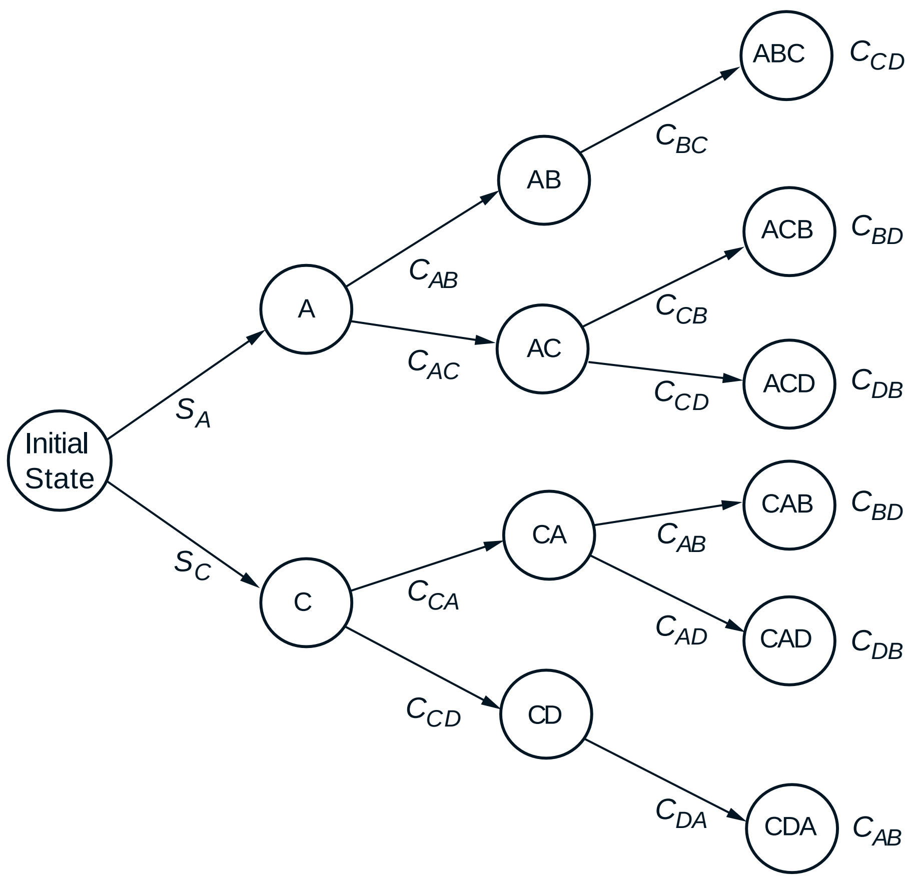
$$u_k \in U_k(x_k)$$

  - w_k is random and generated according to a distribution

$$P_{w_k}(x_k, u_k)$$

DETERMINISTIC FINITE-STATE PROBLEMS

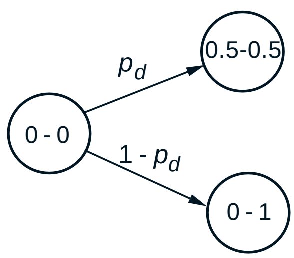
- Scheduling example: Find optimal sequence of operations A, B, C, D
- A must precede B, and C must precede D
- Given startup cost S_A and S_C, and setup transition cost C_{mn} from operation m to operation n



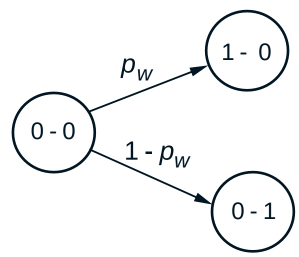
STOCHASTIC FINITE-STATE PROBLEMS

- Example: Find two-game chess match strategy

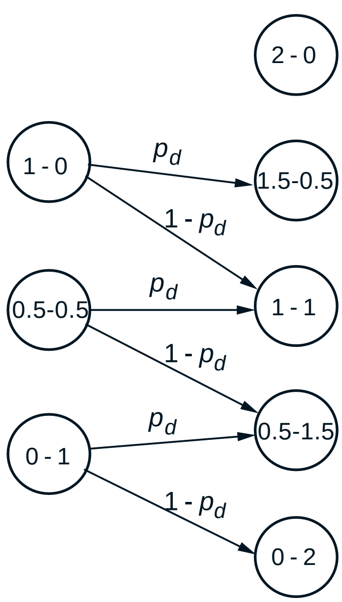
- *Timid* play draws with prob. $p_d > 0$ and loses with prob. $1 - p_d$. *Bold* play wins with prob. $p_w < 1/2$ and loses with prob. $1 - p_w$



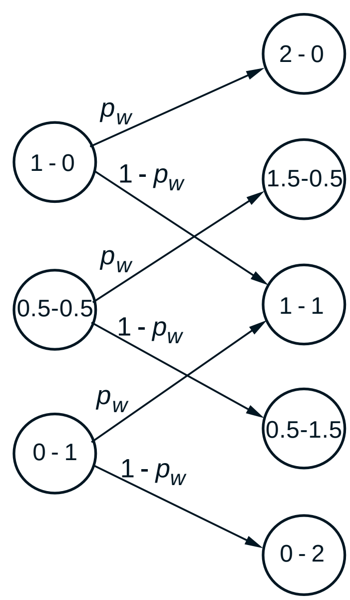
1st Game / Timid Play



1st Game / Bold Play



2nd Game / Timid Play



2nd Game / Bold Play

BASIC PROBLEM

- System $x_{k+1} = f_k(x_k, u_k, w_k)$, $k = 0, \ldots, N-1$
- Control contraints $u_k \in U_k(x_k)$
- Probability distribution $P_k(\cdot \mid x_k, u_k)$ of w_k
- Policies $\pi = \{\mu_0, \ldots, \mu_{N-1}\}$, where μ_k maps states x_k into controls $u_k = \mu_k(x_k)$ and is such that $\mu_k(x_k) \in U_k(x_k)$ for all x_k
- Expected cost of π starting at x_0 is

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

- Optimal cost function

$$J^*(x_0) = \min_\pi J_\pi(x_0)$$

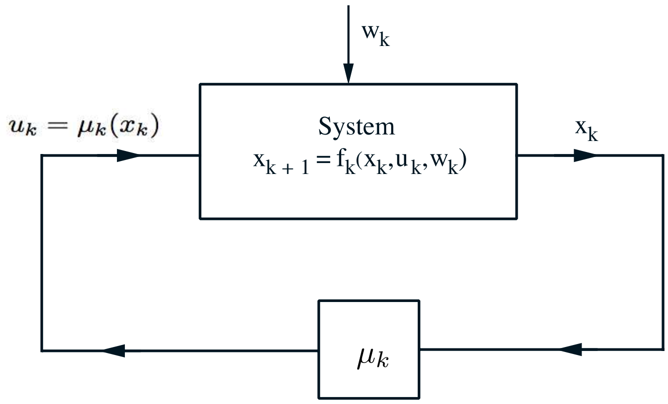
- Optimal policy π^* satisfies

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

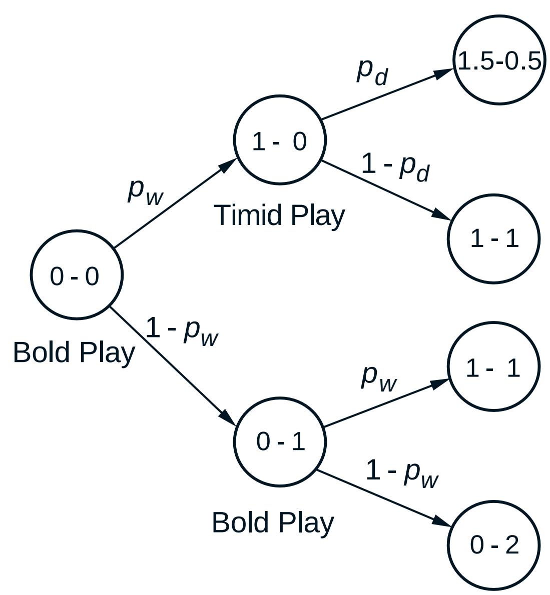
When produced by DP, π^* is independent of x_0.

SIGNIFICANCE OF FEEDBACK

- Open-loop versus closed-loop policies



- In deterministic problems open loop is as good as closed loop
- Value of information; chess match example
- Example of open-loop policy: Play always bold
- Consider the closed-loop policy: Play timid if and only if you are ahead



VARIANTS OF DP PROBLEMS

- Continuous-time problems
- Imperfect state information problems
- Infinite horizon problems
- Suboptimal control

LECTURE BREAKDOWN

- Finite Horizon Problems (Vol. 1, Ch. 1-6)
  - Ch. 1: The DP algorithm (2 lectures)
  - Ch. 2: Deterministic finite-state problems (1 lecture)
  - Ch. 4: Stochastic DP problems (2 lectures)
  - Ch. 5: Imperfect state information problems (2 lectures)
  - Ch. 6: Suboptimal control (2 lectures)
- Infinite Horizon Problems - Simple (Vol. 1, Ch. 7, 3 lectures)

****

- Infinite Horizon Problems - Advanced (Vol. 2)
  - Chs. 1, 2: Discounted problems - Computational methods (3 lectures)
  - Ch. 3: Stochastic shortest path problems (2 lectures)
  - Chs. 6, 7: Approximate DP (6 lectures)

COURSE ADMINISTRATION

- Homework ... once a week or two weeks (30% of grade)

- In class midterm, near end of October ... will cover finite horizon and simple infinite horizon material (30% of grade)

- Project (40% of grade)

- Collaboration in homework allowed but individual solutions are expected

- Prerequisites: Introductory probability, good gasp of advanced calculus (including convergence concepts)

- Textbook: Vol. I of text is required. Vol. II is strongly recommended, but you may be able to get by without it using OCW material (including videos)

A NOTE ON THESE SLIDES

- These slides are a teaching aid, not a text
- Don't expect a rigorous mathematical development or precise mathematical statements
- Figures are meant to convey and enhance ideas, not to express them precisely
- Omitted proofs and a much fuller discussion can be found in the textbook, which these slides follow

MIT OpenCourseWare
http://ocw.mit.edu

6.231 Dynamic Programming and Stochastic Control
Fall 2015

For information about citing these materials or our Terms of Use, visit: http://ocw.mit.edu/terms.