Topics in Reinforcement Learning: Rollout and Approximate Policy Iteration

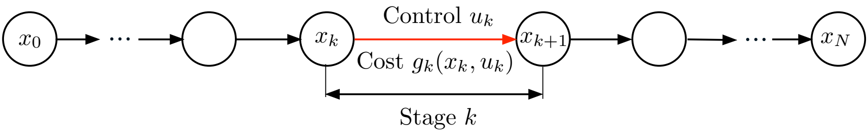
ASU, CSE 691, Spring 2021

Links to Class Notes, Videolectures, and Slides at
http://web.mit.edu/dimitrib/www/RLbook.html

Dimitri P. Bertsekas
dbertsek@asu.edu

Lecture 2
Stochastic Finite and Infinite Horizon DP

Review - Finite Horizon Deterministic Problem



- System

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

where x_k: State, u_k: Control chosen from some set $U_k(x_k)$

- Cost function:

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

- For given initial state x_0, minimize over control sequences $\{u_0, \ldots, u_{N-1}\}$

$$J(x_0; u_0, \ldots, u_{N-1}) = g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k)$$

- Optimal cost function $J^*(x_0) = \min_{\substack{u_k \in U_k(x_k) \\ k=0,\ldots,N-1}} J(x_0; u_0, \ldots, u_{N-1})$

Review - DP Algorithm for Deterministic Problems

Go backward to compute the optimal costs $J_k^*(x_k)$ of the x_k-tail subproblems

Start with

$$J_N^*(x_N) = g_N(x_N), \qquad \text{for all } x_N,$$

and for $k = 0, \ldots, N-1$, let

$$J_k^*(x_k) = \min_{u_k \in U_k(x_k)} \Big[g_k(x_k, u_k) + J_{k+1}^*\big(f_k(x_k, u_k)\big) \Big], \qquad \text{for all } x_k.$$

Then optimal cost $J^*(x_0)$ is obtained at the last step: $J_0^*(x_0) = J^*(x_0)$.

Go forward to construct optimal control sequence $\{u_0^*, \ldots, u_{N-1}^*\}$

Start with

$$u_0^* \in \arg \min_{u_0 \in U_0(x_0)} \Big[g_0(x_0, u_0) + J_1^*\big(f_0(x_0, u_0)\big) \Big], \qquad x_1^* = f_0(x_0, u_0^*).$$

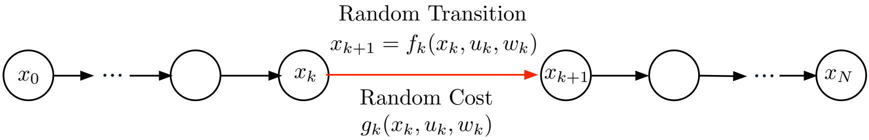
Sequentially, going forward, for $k = 1, 2, \ldots, N-1$, set

$$u_k^* \in \arg \min_{u_k \in U_k(x_k^*)} \Big[g_k(x_k^*, u_k) + J_{k+1}^*\big(f_k(x_k^*, u_k)\big) \Big], \qquad x_{k+1}^* = f_k(x_k^*, u_k^*).$$

Approximation in value space approach: We replace J_k^* with an approximation $\tilde{J}_k$.

Outline

Stochastic DP Problems - Perfect State Observation (We Know x_k)



- System $x_{k+1} = f_k(x_k, u_k, w_k)$ with random "disturbance" w_k (e.g., physical noise, market uncertainties, demand for inventory, unpredictable breakdowns, etc)
- Cost function:

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

- Policies $\pi = \{\mu_0, \dots, \mu_{N-1}\}$, where μ_k is a "closed-loop control law" or "feedback policy"/a function of x_k. A "lookup table" for the control $u_k = \mu_k(x_k)$ to apply at x_k.
- For given initial state x_0, minimize over all $\pi = \{\mu_0, \dots, \mu_{N-1}\}$ the cost

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

- Optimal cost function: $J^*(x_0) = \min_\pi J_\pi(x_0)$. Optimal policy: $J_{\pi^*}(x_0) = J^*(x_0)$

The Stochastic DP Algorithm

Produces the optimal costs $J_k^*(x_k)$ of the tail subproblems that start at x_k

Start with $J_N^*(x_N) = g_N(x_N)$, and for $k = 0, \dots, N-1$, let

$$J_k^*(x_k) = \min_{u_k \in U_k(x_k)} E_{w_k}\Big\{ g_k(x_k, u_k, w_k) + J_{k+1}^*\big(f_k(x_k, u_k, w_k)\big) \Big\}, \qquad \text{for all } x_k.$$

- The optimal cost $J^*(x_0)$ is obtained at the last step: $J_0^*(x_0) = J^*(x_0)$.
- The optimal policy component μ_k^* can be constructed simultaneously with J_k^*, and consists of the minimizing $u_k^* = \mu_k^*(x_k)$ above.

Alternative on-line implementation of the optimal policy, given $J_1^*, \dots, J_{N-1}^*$

Sequentially, going forward, for $k = 0, 1, \dots, N-1$, observe x_k and apply

$$u_k^* \in \arg\min_{u_k \in U_k(x_k)} E_{w_k}\Big\{ g_k(x_k, u_k, w_k) + J_{k+1}^*\big(f_k(x_k, u_k, w_k)\big) \Big\}.$$

Issues: Need to know J_{k+1}^*, compute expectation for each u_k, minimize over all u_k

Approximation in value space: Use $\tilde{J}_k$ in place of J_k^*; approximate $E\{\cdot\}$ and $\min_{u_k}$.

Approximation in Value Space - The Three Approximations

Simplified minimization

First Step

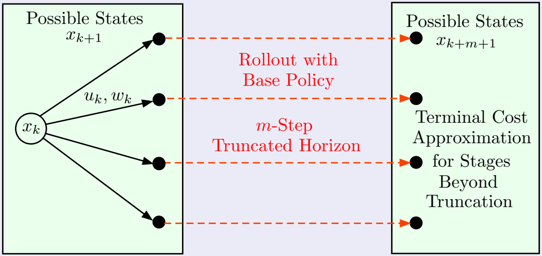
"Future"

At x_k $\longrightarrow$ $$\min_{u_k} E\Big\{\underbrace{g_k(x_k, u_k, w_k)}_{\text{First Step}} + \underbrace{\tilde{J}_{k+1}(x_{k+1})}_{\text{"Future"}}\Big\}$$ "On-Line Play"

Expected value approximation **Cost-to-go approximation**

Important variants: Use multistep lookahead, replace $E\{\cdot\}$ by limited simulation (e.g., a "certainty equivalent" of w_k), multiagent rollout (for multicomponent control problems)

An example: Truncated rollout with base policy and terminal cost approximation (however obtained)



DP Algorithm and Value Space Approximation for Q-Factors

- Optimal Q-factors are given by

$$Q_k^*(x_k, u_k) = E_{w_k}\Big\{g_k(x_k, u_k, w_k) + J_{k+1}^*\big(f_k(x_k, u_k, w_k)\big)\Big\}$$

They define optimal cost-to-go functions and optimal policies by

$$J_k^*(x_k) = \min_{u_k \in U_k(x_k)} Q_k^*(x_k, u_k), \qquad \mu_k^*(x_k) \in \arg\min_{u_k \in U_k(x_k)} Q_k^*(x_k, u_k)$$

- DP algorithm can be written in terms of Q-factors

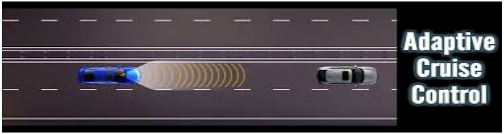
$$Q_k^*(x_k, u_k) = E_{w_k}\Big\{g_k(x_k, u_k, w_k) + \min_{u_{k+1}} Q_{k+1}^*\big(f_k(x_k, u_k, w_k), u_{k+1}\big)\Big\}$$

- Approximately optimal Q-factors $\tilde{Q}_k(x_k, u_k)$, define suboptimal cost-to-go functions and suboptimal policies by

$$\tilde{J}_k(x_k) = \min_{u_k \in U_k(x_k)} \tilde{Q}_k(x_k, u_k), \qquad \tilde{\mu}_k(x_k) \in \arg\min_{u_k \in U_k(x_k)} \tilde{Q}_k(x_k, u_k)$$

- There are many methods to compute $\tilde{Q}_k(x_k, u_k)$, including NN training
- $\tilde{Q}_k$ or $\tilde{J}_k$? An important tradeoff: On-line min simplification vs on-line replanning

A Very Favorable Case: Linear-Quadratic Problems



An example of a linear-quadratic problem

- Keep car velocity constant (like oversimplified cruise control): $x_{k+1} = x_k + bu_k + w_k$
- u_k is unconstrained; w_k has 0-mean and variance σ^2
- Here $x_k = v_k - \bar{v}$ is the deviation between the vehicle's velocity v_k at time k from desired level $\bar{v}$, and b is given
- Cost over N stages: $x_N^2 + \sum_{k=0}^{N-1}(x_k^2 + ru_k^2)$, where $r \geq 0$ is given
- DP algorithm:

$$J_N^*(x_N) = x_N^2,$$

$$J_k^*(x_k) = \min_{u_k} E_{w_k}\left\{x_k^2 + ru_k^2 + J_{k+1}^*(x_k + bu_k + w_k)\right\}, \quad k = 0, \ldots, N-1$$

- DP algorithm can be carried out in closed form to yield
  $J_k^*(x_k) = K_k x_k^2 + \text{const}, \;\; \mu_k^*(x_k) = L_k x_k$: K_k and L_k can be explicitly computed
- The solution does not depend on the distribution of w_k as long as it has 0 mean: Certainty Equivalence (a common approximation idea for other problems)

Derivation

$$
\begin{aligned}
J^*_{N-1}(x_{N-1}) &= \min_{u_{N-1}} E\left\{x^2_{N-1} + ru^2_{N-1} + J^*_N(x_{N-1} + bu_{N-1} + w_{N-1})\right\} \\
&= \min_{u_{N-1}} E\left\{x^2_{N-1} + ru^2_{N-1} + (x_{N-1} + bu_{N-1} + w_{N-1})^2\right\} \\
&= \min_{u_{N-1}} \left[x^2_{N-1} + ru^2_{N-1} + (x_{N-1} + bu_{N-1})^2 + 2E\{w_{N-1}\}(x_{N-1} + bu_{N-1}) + E\{w^2_{N-1}\}\right] \\
&= x^2_{N-1} + \min_{u_{N-1}} \left[ru^2_{N-1} + (x_{N-1} + bu_{N-1})^2\right] + \sigma^2
\end{aligned}
$$

Minimize by setting to zero the derivative: $0 = 2ru_{N-1} + 2b(x_{N-1} + bu_{N-1})$, to obtain

$$\mu^*_{N-1}(x_{N-1}) = -\frac{b}{r + b^2}\, x_{N-1} = L_{N-1}x_{N-1}$$

and by substitution, $J^*_{N-1}(x_{N-1}) = P_{N-1}x^2_{N-1} + \sigma^2$, where $P_{N-1} = \frac{r}{r+b^2} + 1$

Similarly, going backwards, we obtain for all k:

$$J^*_k(x_k) = P_k x_k^2 + \sigma^2 \sum_{m=k}^{N-1} P_{m+1}, \quad \mu^*_k(x_k) = L_k x_k, \quad P_k = \frac{rP_{k+1}}{r + b^2 P_{k+1}} + 1, \quad L_k = -\frac{bP_{k+1}}{r + b^2 P_{k+1}}$$

Linear-Quadratic Problems in General

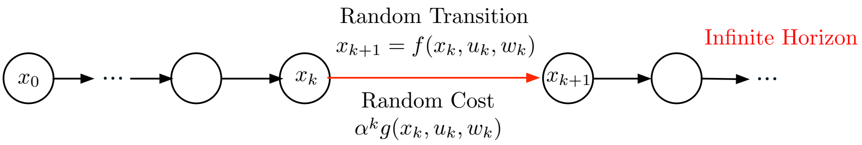
Observations and generalizations

- The solution does not depend on the distribution of w_k, only on the mean, i.e., we have certainty equivalence
- Generalization to multidimensional problems, nonzero mean disturbances, etc
- Generalization to problems where the state is observed partially through linear measurements: Optimal policy involves an extended form of certainty equivalence

$$L_k E\{x_k \mid \text{measurements}\}$$

  where $E\{x_k \mid \text{measurements}\}$ is provided by an estimator (e.g., Kalman filter)
- Linear systems and quadratic cost are a starting point for other lines of investigations and approximations:
  - Problems with safety/state constraints [Model Predictive Control (MPC)]
  - Problems with control constraints (MPC)
  - Unknown or changing system parameters (adaptive control)

Infinite Horizon Problems



Infinite number of stages, and stationary system and cost

- System $x_{k+1} = f(x_k, u_k, w_k)$ with state, control, and random disturbance
- Policies $\pi = \{\mu_0, \mu_1, \ldots\}$ with $\mu_k(x) \in U(x)$ for all x and k
- Cost of stage k: $\alpha^k g\big(x_k, \mu_k(x_k), w_k\big)$
- Cost of a policy $\pi = \{\mu_0, \mu_1, \ldots\}$: The limit as $N \to \infty$ of the N-stage costs

$$J_\pi(x_0) = \lim_{N\to\infty} E_{w_k}\left\{\sum_{k=0}^{N-1} \alpha^k g\big(x_k, \mu_k(x_k), w_k\big)\right\}$$

- $0 < \alpha \leq 1$ is the discount factor. If $\alpha < 1$ the problem is called discounted
- Optimal cost function $J^*(x_0) = \min_\pi J_\pi(x_0)$
- Problems with $\alpha = 1$ typically include a special cost-free termination state t. The objective is to reach (or approach) t at minimum expected cost.

Finite horizon opt. costs –> Infinite horizon opt. cost: Consider the N-stages problem, with terminal cost 0

- Apply DP, let $V_{N-k}(x)$ be the optimal cost-to-go starting at x with k stages to go:

$$V_{N-k}(x) = \min_{u \in U(x)} E_w\Big\{\alpha^{N-k} g(x,u,w) + V_{N-k+1}\big(f(x,u,w)\big)\Big\}, \quad V_N(x) \equiv 0$$

- Define $J_k(x) = V_{N-k}(x)/\alpha^{N-k}$, i.e., reverse the time index and divide with α^{N-k}:

$$J_k(x) = \min_{u \in U(x)} E_w\Big\{g(x,u,w) + \alpha J_{k-1}\big(f(x,u,w)\big)\Big\}, \quad J_0(x) \equiv 0 \qquad \text{(VI)}$$

- $J_N(x)$ is equal to $V_0(x)$, the N-stages optimal cost starting from x
- So for any k, $J_k(x)$ = k-stages optimal cost starting from x. Intuitively:

$$J^*(x) = \lim_{k \to \infty} J_k(x), \qquad \text{for all states } x \quad (??)$$

J^* satisfies Bellman's equation: Take the limit in Eq. (VI)

$$J^*(x) = \min_{u \in U(x)} E_w\Big\{g(x,u,w) + \alpha J^*\big(f(x,u,w)\big)\Big\}, \qquad \text{for all states } x \quad (??)$$

Optimality condition: Let $\mu^*(x)$ attain the min in the Bellman equation for all x

The policy $\{\mu^*, \mu^*, \ldots\}$ is optimal (??). (This type of policy is called stationary.)

Infinite Horizon Problems - Algorithms

Value iteration (VI): Generates finite horizon opt. cost function sequence $\{J_k\}$

$$J_k(x) = \min_{u \in U(x)} E_w\Big\{g(x,u,w) + \alpha J_{k-1}\big(f(x,u,w)\big)\Big\}, \qquad J_0 \text{ is "arbitrary" (??)}$$

Policy Iteration (PI): Generates sequences of policies $\{\mu^k\}$ and their cost functions $\{J_{\mu^k}\}$; μ^0 is "arbitrary"

The typical iteration starts with a policy μ and generates a new policy $\tilde{\mu}$ in two steps:

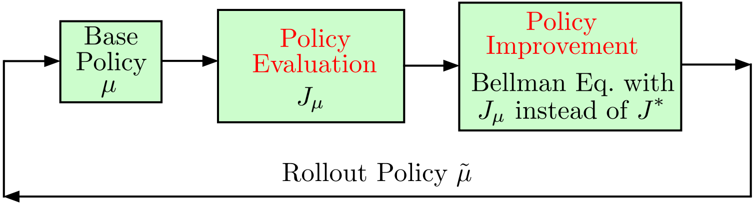
- Policy evaluation step, which computes the cost function J_μ
- Policy improvement step, which computes the improved rollout policy $\tilde{\mu}$ using the one-step lookahead minimization

$$\tilde{\mu}(x) \in \arg\min_{u \in U(x)} E_w\Big\{g(x,u,w) + \alpha J_\mu\big(f(x,u,w)\big)\Big\}$$

There are several options for policy evaluation to compute J_μ

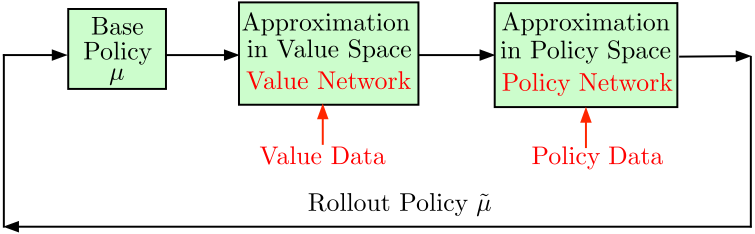
- Solve Bellman's equation for μ [$J_\mu(x) = E\{g(x,\mu(x),w) + \alpha J_\mu(f(x,\mu(x),w))\}$] by using VI or other method (it is linear in J_μ)
- Use simulation (on-line Monte-Carlo, Temporal Difference (TD) methods)

Exact and Approximate Policy Iteration



Important facts (to be discussed later):

- PI yields in the limit an optimal policy (??)
- PI is faster than VI; can be viewed as Newton's method for solving Bellman's Eq.
- PI can be implemented approximately, with a value and (perhaps) a policy network



Example - Linear Quadratic Problem for Infinite Horizon

- System $x_{k+1} = x_k + bu_k + w_k$ and cost function

$$\lim_{N\to\infty} E\left\{\sum_{k=0}^{N-1} \alpha^k (x_k^2 + ru_k^2)\right\}$$

- The VI algorithm is

$$J_{k+1}(x) = \min_u E_w\big\{x^2 + ru^2 + \alpha J_k(x + bu + w)\big\}$$

- Similar to the finite horizon case, the value iterates J_k are quadratic:

$$J_0(x) = 0, \qquad J_{k+1}(x) = K_k x^2 + \text{constant} \cdot \sigma^2,$$

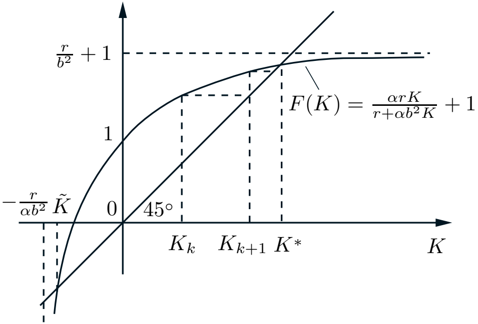
  where $\{K_k\}$ is generated by

$$K_0 = 1, \qquad K_{k+1} = \frac{\alpha r K_k}{r + \alpha b^2 K_k} + 1$$

- It can be shown that $\{K_k\}$ converges to a limit K^* for any $K_0 \geq 0$; see the next slide
- The function $J^*(x) = K^* x^2 + \text{constant}$ is the solution of Bellman's equation
- The optimal policy is a linear function of x, $\mu^*(x) = Lx$, and is obtained from

$$\mu^*(x) \in \arg\min_u E_w\big\{x^2 + ru^2 + \alpha K^*(x + bu + w)^2\big\}$$

Linear Quadratic Problem Solution - Geometric Interpretation



- The Bellman equation (neglecting the constant, i.e. $w \equiv 0$) is written as

$$K^* x^2 = \min_u \left[x^2 + ru^2 + \alpha K^*(x+bu)^2\right] = F(K^*)x^2,$$

where

$$F(K) = \frac{\alpha r K}{r + \alpha b^2 K} + 1$$

- So $K^* = F(K^*)$, i.e., K^* is a fixed point of the function F
- VI algorithm is $J_{k+1}(x) = K_{k+1}x^2 = F(K_k)x^2$
- Cancelling x^2, VI is equivalent to the fixed point iteration $K_{k+1} = F(K_k)$

Example - Policy Iteration for the Linear Quadratic Problem

Starts with linear policy $\mu^0(x) = L_0 x$, generates sequence of linear policies $\mu^k(x) = L_k x$ (see class notes for details)

- Policy evaluation:

$$J_{\mu^k}(x) = K_k x^2 + \text{constant}$$

where

$$K_k = \frac{1 + rL_k^2}{1 - \alpha(1 + bL_k)^2}$$

- Policy improvement:

$$\mu^{k+1}(x) = L_{k+1} x$$

where

$$L_{k+1} = -\frac{\alpha b K_k}{r + \alpha b^2 K_k}$$

- Can be viewed as Newton's method for solving the Riccati equation

$$K = \frac{\alpha r K}{r + \alpha b^2 K} + 1$$

- Rollout is a single Newton iteration

A More Abstract View of VI and PI

Bellman's equation, VI, and PI can be written using Bellman operators

Recall Bellman's equation

$$J^*(x) = \min_{u \in U(x)} E_w \Big\{ g(x,u,w) + \alpha J^*\big(f(x,u,w)\big) \Big\}, \qquad \text{for all states } x$$

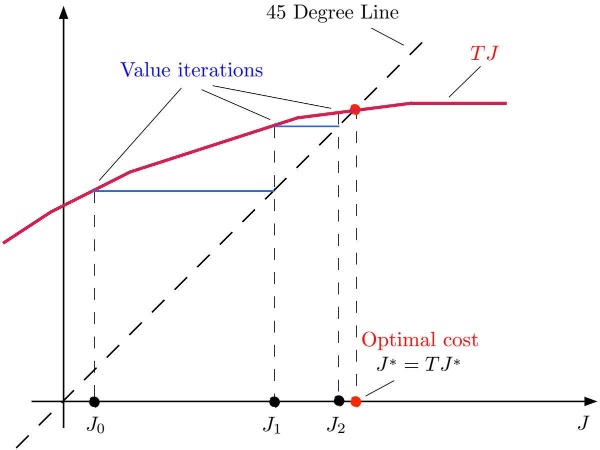
It can be written as a fixed point equation: $J^*(x) = (TJ^*)(x)$, where T is the Bellman operator that transforms a function $J(\cdot)$ into a function $(TJ)(\cdot)$

$$(TJ)(x) = \min_{u \in U(x)} E_w \Big\{ g(x,u,w) + \alpha J\big(f(x,u,w)\big) \Big\}, \qquad \text{for all states } x$$

Shorthand theory using Bellman operators:

- VI is the fixed point iteration $J_{k+1} = TJ_k$
- There is a Bellman operator T_μ for any policy μ and corresponding Bellman Eq. $J_\mu(x) = (T_\mu J_\mu)(x) = E\{g(x,\mu(x),w) + \alpha J_\mu(f(x,\mu(x),w))\}$
- PI is written compactly as $J_{\mu^k} = T_{\mu^k} J_{\mu^k}$ (policy evaluation) and $T_{\mu^{k+1}} J_{\mu^k} = TJ_{\mu^k}$ (policy improvement)
- The PI sequence $\{J_{\mu^k}\}$ is the result of Newton's method for solving $J = TJ$

Value Iteration - Geometric Interpretation (Spend Time to Understand)

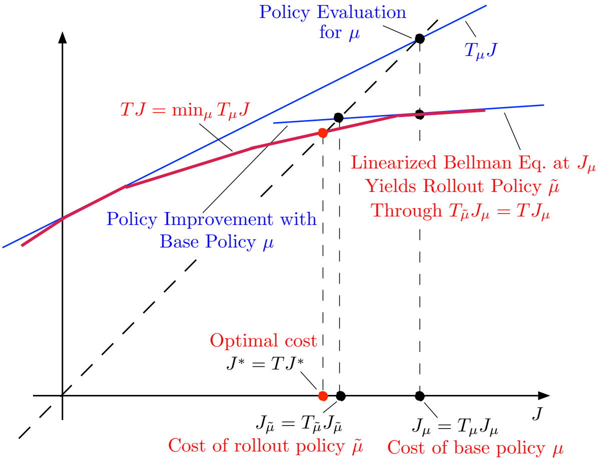


Value iteration:

$$J_{k+1}(x) = (TJ_k)(x) = \min_{u \in U(x)} E_w \Big\{ g(x,u,w) + \alpha J_k\big(f(x,u,w)\big) \Big\}$$

where T is the Bellman operator that maps functions $J(\cdot)$ to functions $(TJ)(\cdot)$

Policy Iteration - Geometric Interpretation (Spend Time to Understand)



Given the current policy μ:

- The rollout policy is obtained by $J_\mu = T_\mu J_\mu$ (policy evaluation) and $T_{\tilde{\mu}} J_\mu = T J_\mu$ (policy improvement)
- The rollout algorithm is a single iteration of PI/Newton's method

About the Next Lecture

We will cover problem formulations and reformulations

- How do we formulate DP models for practical problems?
- Problems involving a terminal state (stochastic shortest path problems)
- Problem reformulation by state augmentation (dealing with delays, correlations, forecasts, etc)
- Problems involving imperfect state observation (POMDP or Partial Observation MDP)
- Multiagent problems - Nonclassical information patterns
- Systems with unknown or changing parameters - Adaptive control

PLEASE READ AS MUCH OF SECTION 1.4 OF THE CLASS NOTES AS YOU CAN

1ST HOMEWORK (DUE IN ONE WEEK) TO BE ANNOUNCED ON-LINE