6.231 DYNAMIC PROGRAMMING

LECTURE 9

LECTURE OUTLINE

- Rollout algorithms
- Policy improvement property
- Discrete deterministic problems
- Approximations of rollout algorithms
- Model Predictive Control (MPC)
- Discretization of continuous time
- Discretization of continuous space
- Other suboptimal approaches

ROLLOUT ALGORITHMS

- One-step lookahead policy: At each k and state x_k, use the control $\overline{\mu}_k(x_k)$ that

$$\min_{u_k \in U_k(x_k)} E\Big\{g_k(x_k, u_k, w_k) + \tilde{J}_{k+1}\big(f_k(x_k, u_k, w_k)\big)\Big\},$$

where

  - $\tilde{J}_N = g_N$.
  - $\tilde{J}_{k+1}$: approximation to true cost-to-go J_{k+1}

- Rollout algorithm: When $\tilde{J}_k$ is the cost-to-go of some heuristic policy (called the base policy)

- Policy improvement property (to be shown): The rollout algorithm achieves no worse (and usually much better) cost than the base heuristic starting from the same state.

- Main difficulty: Calculating $\tilde{J}_k(x_k)$ may be computationally intensive if the cost-to-go of the base policy cannot be analytically calculated.
  - May involve Monte Carlo simulation if the problem is stochastic.
  - Things improve in the deterministic case.

EXAMPLE: THE QUIZ PROBLEM

- A person is given N questions; answering correctly question i has probability p_i, reward v_i. Quiz terminates at the first incorrect answer.
- Problem: Choose the ordering of questions so as to maximize the total expected reward.
- Assuming no other constraints, it is optimal to use the index policy: Answer questions in decreasing order of $p_i v_i/(1-p_i)$.
- With minor changes in the problem, the index policy need not be optimal. Examples:
  - A limit ($< N$) on the maximum number of questions that can be answered.
  - Time windows, sequence-dependent rewards, precedence constraints.
- Rollout with the index policy as base policy: Convenient because at a given state (subset of questions already answered), the index policy and its expected reward can be easily calculated.
- Very effective for solving the quiz problem and important generalizations in scheduling (see Bertsekas and Castanon, J. of Heuristics, Vol. 5, 1999).

COST IMPROVEMENT PROPERTY

- Let

$\overline{J}_k(x_k)$: Cost-to-go of the rollout policy

$H_k(x_k)$: Cost-to-go of the base policy

- We claim that $\overline{J}_k(x_k) \le H_k(x_k)$ for all x_k, k

- Proof by induction: We have $\overline{J}_N(x_N) = H_N(x_N)$ for all x_N. Assume that

$$\overline{J}_{k+1}(x_{k+1}) \le H_{k+1}(x_{k+1}), \quad \forall\, x_{k+1}.$$

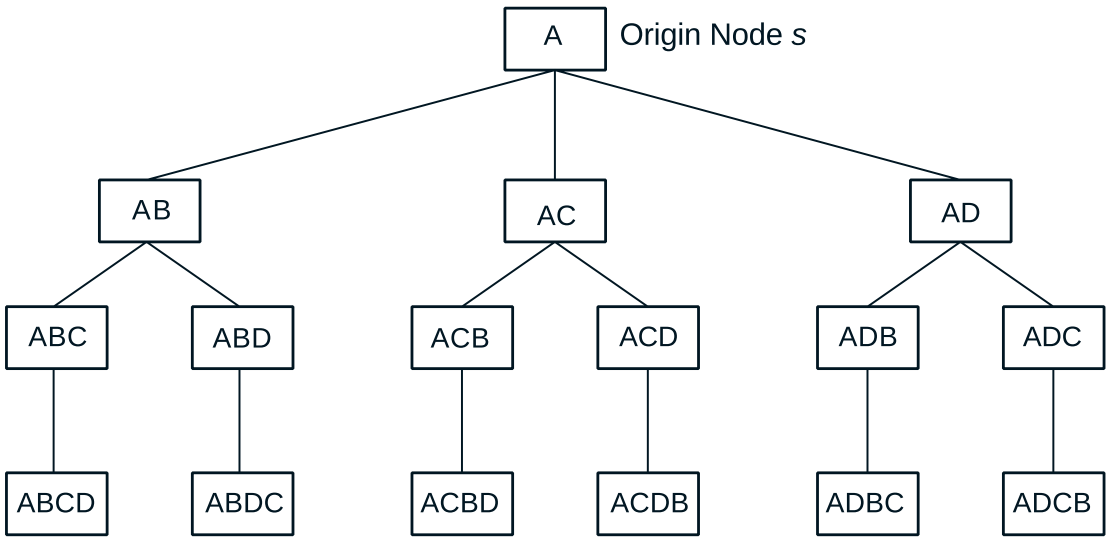
Let $\overline{\mu}_k(x_k)$ and $\mu_k(x_k)$ be the controls applied by rollout and heuristic at x_k. Then, for all x_k

$$\begin{aligned}
\overline{J}_k(x_k) &= E\Big\{g_k\big(x_k,\overline{\mu}_k(x_k),w_k\big) + \overline{J}_{k+1}\Big(f_k\big(x_k,\overline{\mu}_k(x_k),w_k\big)\Big)\Big\}\\
&\le E\Big\{g_k\big(x_k,\overline{\mu}_k(x_k),w_k\big) + H_{k+1}\Big(f_k\big(x_k,\overline{\mu}_k(x_k),w_k\big)\Big)\Big\}\\
&\le E\Big\{g_k\big(x_k,\mu_k(x_k),w_k\big) + H_{k+1}\Big(f_k\big(x_k,\mu_k(x_k),w_k\big)\Big)\Big\}\\
&= H_k(x_k)
\end{aligned}$$

- Induction hypothesis ==> 1st inequality
- Min selection of $\overline{\mu}_k(x_k)$ ==> 2nd inequality
- Definition of H_k, μ_k ==> last equality

DISCRETE DETERMINISTIC PROBLEMS

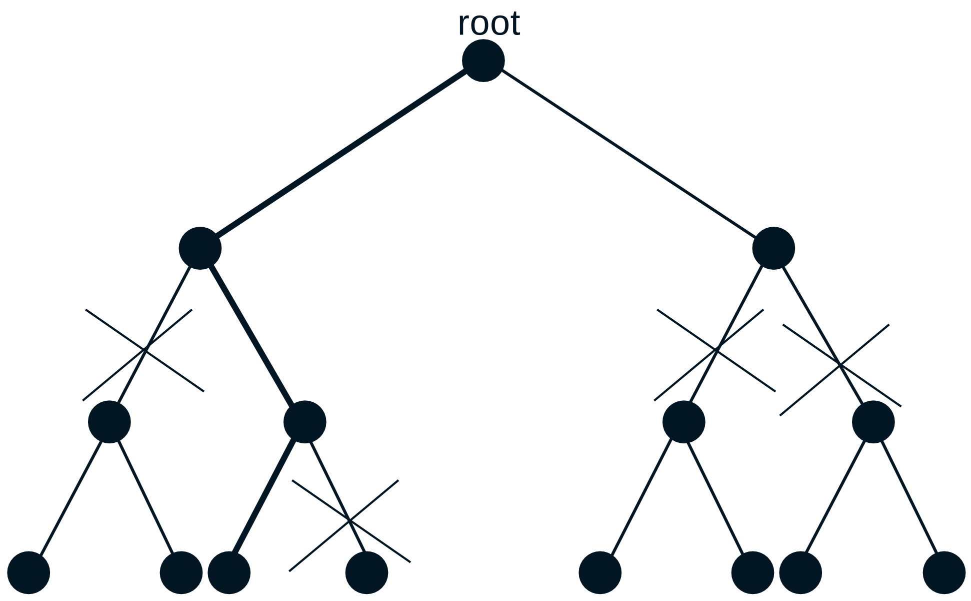
- Any discrete optimization problem can be represented sequentially by breaking down the decision process into stages.
- A tree/shortest path representation. The leaves of the tree correspond to the feasible solutions.
- Example: Traveling salesman problem. Find a minimum cost tour through N cities.



Traveling salesman problem with four cities A, B, C, D

- Complete partial solutions, one stage at a time
- May apply rollout with any heuristic that can complete a partial solution
- No costly stochastic simulation needed

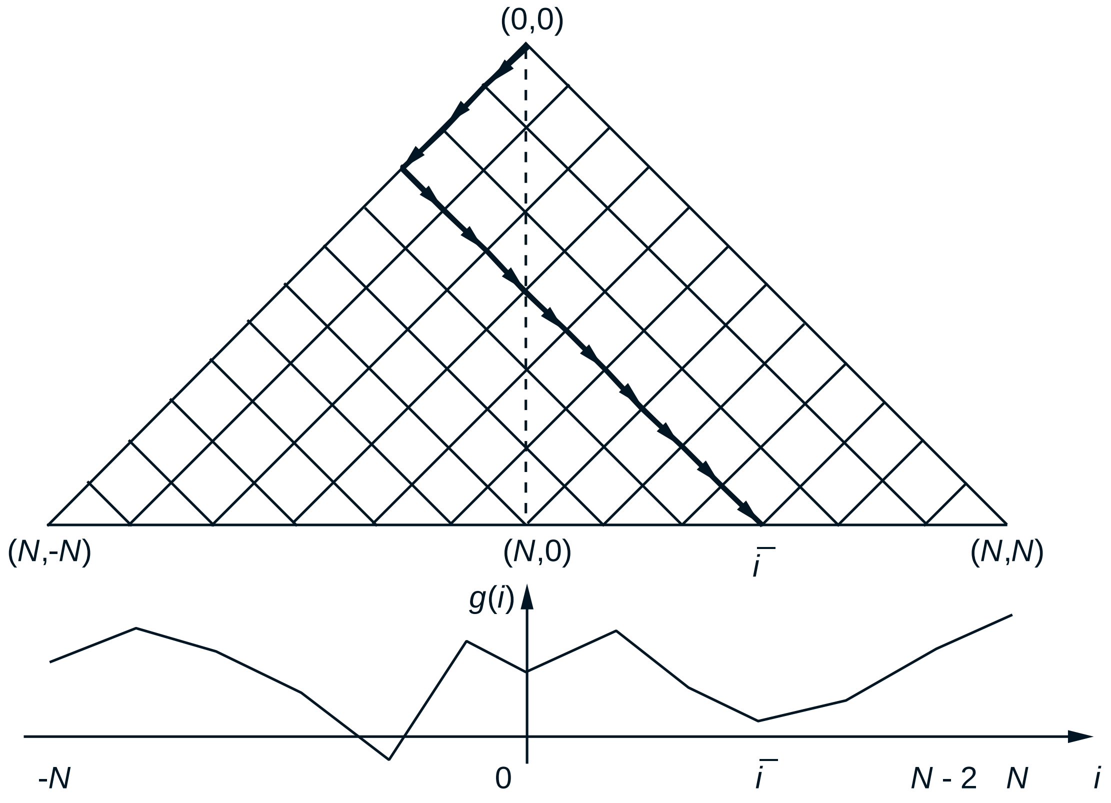
EXAMPLE: THE BREAKTHROUGH PROBLEM



- Given a binary tree with N stages.
- Each arc is free or is blocked (crossed out)
- Problem: Find a free path from the root to the leaves (such as the one shown with thick lines).
- Base heuristic (greedy): Follow the right branch if free; else follow the left branch if free.
- This is a rare rollout instance that admits a detailed analysis.
- For large N and given prob. of free branch: the rollout algorithm requires $O(N)$ times more computation, but has $O(N)$ times larger prob. of finding a free path than the greedy algorithm.

DET. EXAMPLE: ONE-DIMENSIONAL WALK

- A person takes either a unit step to the left or a unit step to the right. Minimize the cost $g(i)$ of the point i where he will end up after N steps.



- Base heuristic: Always go to the right. Rollout finds the rightmost local minimum.

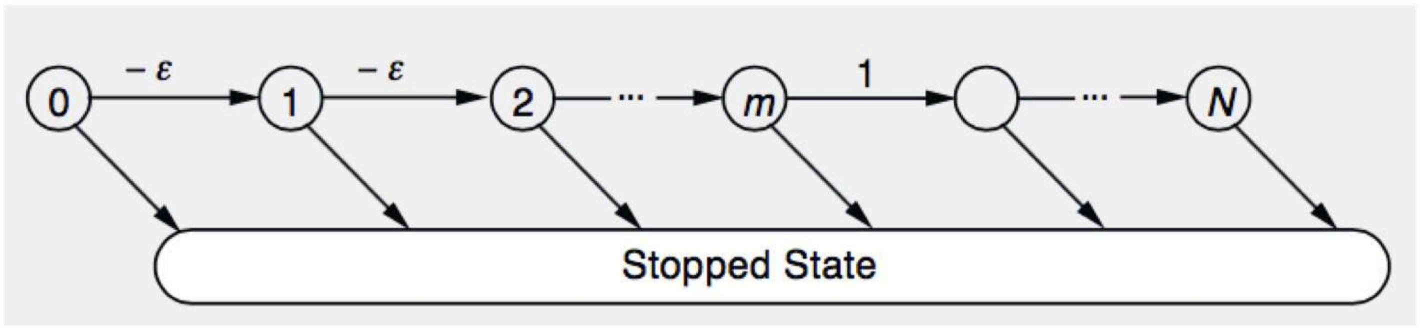
- Alternative base heuristic: Compare always go to the right and always go the left. Choose the best of the two. Rollout finds a global minimum.

A ROLLOUT ISSUE FOR DISCRETE PROBLEMS

- The base heuristic need not constitute a policy in the DP sense.
- Reason: Depending on its starting point, the base heuristic may not apply the same control at the same state.
- As a result the cost improvement property may be lost (except if the base heuristic has a property called sequential consistency; see the text for a formal definition).
- The cost improvement property is restored in two ways:
  - The base heuristic has a property called sequential improvement which guarantees cost reduction at each step (see the text for a formal definition).
  - A variant of the rollout algorithm, called fortified rollout, is used, which enforces cost improvement. Roughly speaking the "best" solution found so far is maintained, and it is followed whenever at any time the standard version of the algorithm tries to follow a "worse" solution (see the text).

ROLLING HORIZON WITH ROLLOUT

- We can use a rolling horizon approximation in calculating the cost-to-go of the base heuristic.
- Because the heuristic is suboptimal, the rationale for a long rolling horizon becomes weaker.
- Example: N-stage stopping problem where the stopping cost is 0, the continuation cost is either $-\epsilon$ or 1, where $0 < \epsilon << 1$, and the first state with continuation cost equal to 1 is state m. Then the optimal policy is to stop at state m, and the optimal cost is $-m\epsilon$.



- Consider the heuristic that continues at every state, and the rollout policy that is based on this heuristic, with a rolling horizon of $\ell \leq m$ steps.
- It will continue up to the first $m - \ell + 1$ stages, thus compiling a cost of $-(m - \ell + 1)\epsilon$. The rollout performance improves as l becomes shorter!
- Limited vision may work to our advantage!

MODEL PREDICTIVE CONTROL (MPC)

- Special case of rollout for linear deterministic systems (similar extensions to nonlinear/stochastic)
  - System: $x_{k+1} = Ax_k + Bu_k$
  - Quadratic cost per stage: $x_k' Q x_k + u_k' R u_k$
  - Constraints: $x_k \in X$, $u_k \in U(x_k)$
- Assumption: For any $x_0 \in X$ there is a feasible state-control sequence that brings the system to 0 in m steps, i.e., $x_m = 0$
- MPC at state x_k solves an m-step optimal control problem with constraint $x_{k+m} = 0$, i.e., finds a sequence $\bar{u}_k, \ldots, \bar{u}_{k+m-1}$ that minimizes

$$\sum_{\ell=0}^{m-1} \left(x_{k+\ell}' Q x_{k+\ell} + u_{k+\ell}' R u_{k+\ell} \right)$$

subject to $x_{k+m} = 0$

- Then applies the first control $\bar{u}_k$ (and repeats at the next state x_{k+1})
- MPC is rollout with heuristic derived from the corresponding $m-1$-step optimal control problem
- Key Property of MPC: Since the heuristic is stable, the rollout is also stable (suggested by policy improvement property; see the text).

DISCRETIZATION

- If the time, and/or state space, and/or control space are continuous, they must be discretized.

- Consistency issue, i.e., as the discretization becomes finer, the cost-to-go functions of the discretized problem should converge to those of the original problem.

- Pitfall with discretizing continuous time: The control constraint set may change a lot as we pass to the discrete-time approximation.

- Example: Consider the system $\dot{x}(t) = u(t)$, with control constraint $u(t) \in \{-1, 1\}$. The reachable states after time δ are $x(t + \delta) = x(t) + u$, with $u \in [-\delta, \delta]$.

- Compare it with the reachable states after we discretize the system naively: $x(t+\delta) = x(t)+\delta u(t)$, with $u(t) \in \{-1, 1\}$.

- "Convexification effect" of continuous time: a discrete control constraint set in continuous-time differential systems, is equivalent to a continuous control constraint set when the system is looked at discrete times.

SPACE DISCRETIZATION

- Given a discrete-time system with state space S, consider a finite subset $\overline{S}$; for example $\overline{S}$ could be a finite grid within a continuous state space S.
- Difficulty: $f(x,u,w) \notin \overline{S}$ for $x \in \overline{S}$.
- We define an approximation to the original problem, with state space $\overline{S}$, as follows:
- Express each $x \in S$ as a convex combination of states in $\overline{S}$, i.e.,

$$x = \sum_{x_i \in \overline{S}} \phi_i(x) x_i \quad \text{where } \phi_i(x) \geq 0, \ \sum_i \phi_i(x) = 1$$

- Define a "reduced" dynamic system with state space $\overline{S}$, whereby from each $x_i \in \overline{S}$ we move to $\overline{x} = f(x_i,u,w)$ according to the system equation of the original problem, and then move to $x_j \in \overline{S}$ with probabilities $\phi_j(\overline{x})$.
- Define similarly the corresponding cost per stage of the transitions of the reduced system.
- Note application to finite-state POMDP (discretization of the simplex of the belief states).

SPACE DISCRETIZATION/AGGREGATION

- Let $\overline{J}_k(x_i)$ be the optimal cost-to-go of the "reduced" problem from each state $x_i \in \overline{S}$ and time k onward.

- Approximate the optimal cost-to-go of any $x \in S$ for the original problem by

$$\tilde{J}_k(x) = \sum_{x_i \in \overline{S}} \phi_i(x) \overline{J}_k(x_i),$$

and use one-step-lookahead based on $\tilde{J}_k$.

- The coefficients $\phi_i(x)$ can be viewed as features in an aggregation scheme.

- Important question: Consistency, i.e., as the number of states in $\overline{S}$ increases, $\tilde{J}_k(x)$ should converge to the optimal cost-to-go of the original prob.

- Interesting observation: While the original problem may be deterministic, the reduced problem is always stochastic.

- Generalization: The set $\overline{S}$ may be any finite set (not a subset of S) as long as the coefficients $\phi_i(x)$ admit a meaningful interpretation that quantifies the degree of association of x with x_i (a form of aggregation).

OTHER SUBOPTIMAL APPROACHES

- Minimize the DP equation error (Fitted Value Iteration): Approximate $J_k(x_k)$ with $\tilde{J}_k(x_k, r_k)$, where r_k is a parameter vector, chosen to minimize some form of error in the DP equations
  - Can be done sequentially going backwards in time (approximate J_k using an approximation of J_{k+1}, starting with $\tilde{J}_N = g_N$).
- Direct approximation of control policies: For a subset of states x^i, $i = 1, \ldots, m$, find

$$\hat{\mu}_k(x^i) = \arg \min_{u_k \in U_k(x^i)} E\Big\{g(x^i, u_k, w_k) \\ + \tilde{J}_{k+1}\big(f_k(x^i, u_k, w_k), r_{k+1}\big)\Big\}$$

Then find $\tilde{\mu}_k(x_k, s_k)$, where s_k is a vector of parameters obtained by solving the problem

$$\min_s \sum_{i=1}^{m} \|\hat{\mu}_k(x^i) - \tilde{\mu}_k(x^i, s)\|^2$$

- Approximation in policy space: Do not bother with cost-to-go approximations. Parametrize the policies as $\tilde{\mu}_k(x_k, s_k)$, and minimize the cost function of the problem over the parameters s_k (random search is a possibility).