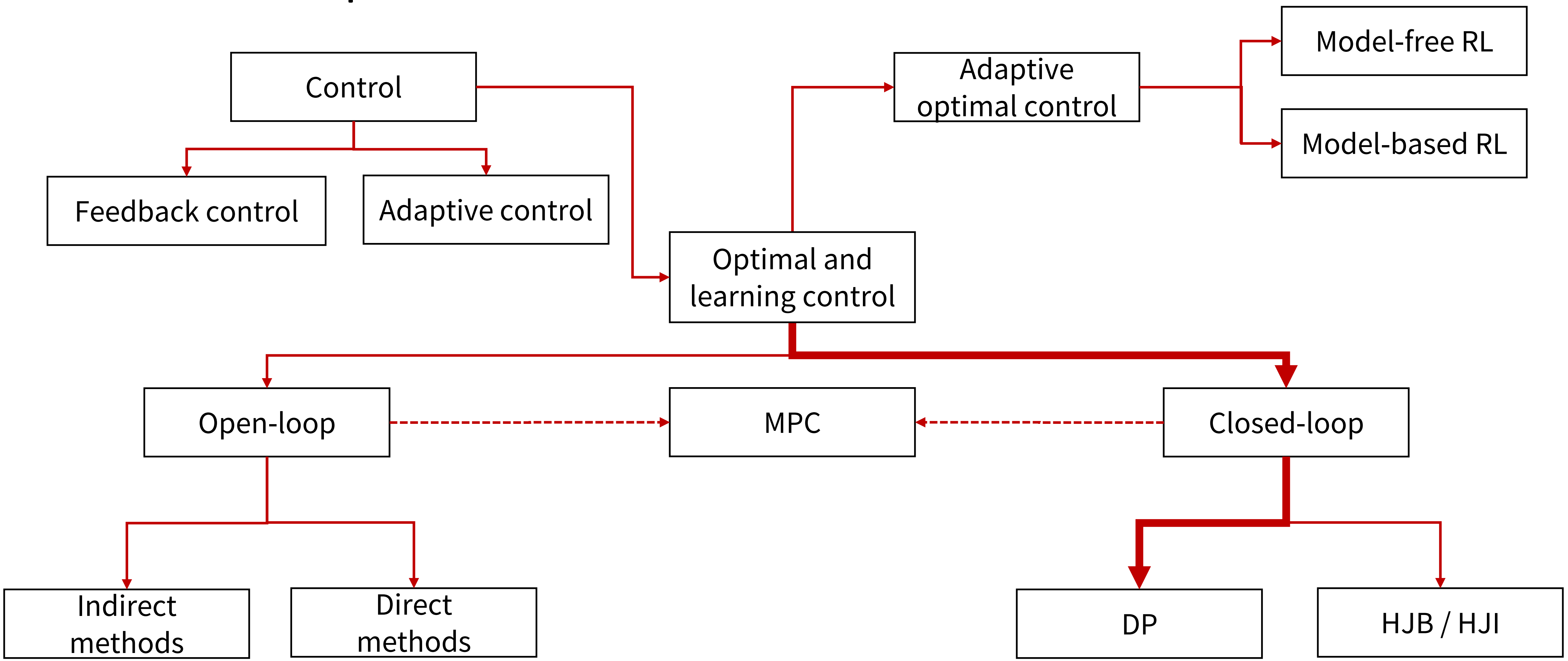
AA203
Optimal and Learning-based Control

Intro to dynamic programming

Stanford University

Autonomous Systems Lab

Roadmap

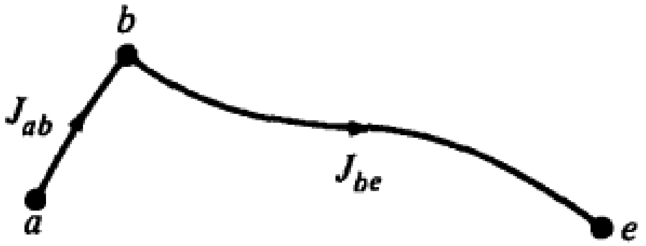


Control
Feedback control
Adaptive control
Optimal and learning control
Adaptive optimal control
Model-free RL
Model-based RL
Open-loop
MPC
Closed-loop
Indirect methods
Direct methods
DP
HJB / HJI

Principle of optimality

The key concept behind the dynamic programming approach is the principle of optimality

Suppose optimal path for a multi-stage decision-making problem is



- first decision yields segment $a - b$ with cost J_{ab}
- remaining decisions yield segments $b - e$ with cost J_{be}
- optimal cost is then $J^*_{ae} = J_{ab} + J_{be}$

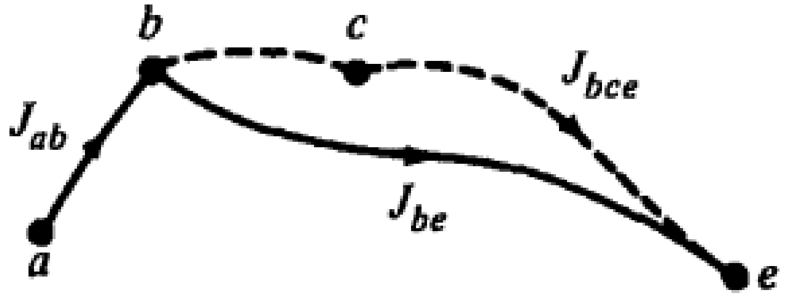
Principle of optimality

- Claim: If $a - b - e$ is optimal path from a to e, then $b - e$ is optimal path from b to e
- *Proof*: Suppose $b - c - e$ is the optimal path from b to e. Then

$$J_{bce} < J_{be}$$

and

$$J_{ab} + J_{bce} < J_{ab} + J_{be} = J_{ae}^*$$



Contradiction!

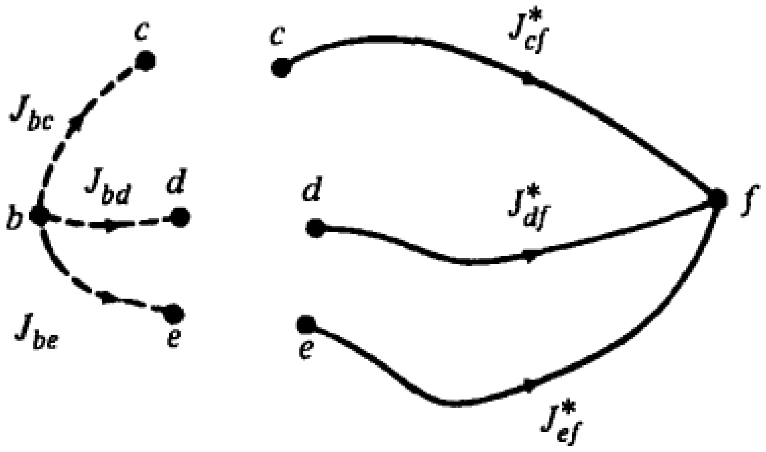
Principle of optimality

Principle of optimality (for discrete-time systems): Let $\pi^* := \{\pi_0^*, \pi_1^*, \ldots, \pi_{N-1}^*\}$ be an optimal policy. Assume state $\mathbf{x}_k$ is reachable. Consider the subproblem whereby we are at $\mathbf{x}_k$ at time k and we wish to minimize the cost-to-go from time k to time N. Then the truncated policy $\{\pi_k^*, \pi_{k+1}^*, \ldots, \pi_{N-1}^*\}$ is optimal for the subproblem

- tail policies optimal for tail subproblems
- notation: $\pi_k^*(\mathbf{x}_k) = \pi^*(\mathbf{x}_k, k)$

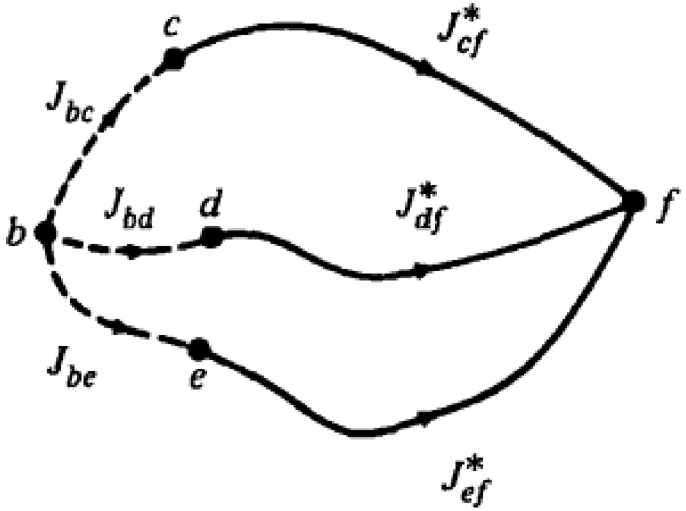
Applying the principle of optimality

Principle of optimality: if $b - c$ is the initial segment of the optimal path from b to f, then $c - f$ is the terminal segment of this path



Hence, the optimal trajectory is found by comparing:

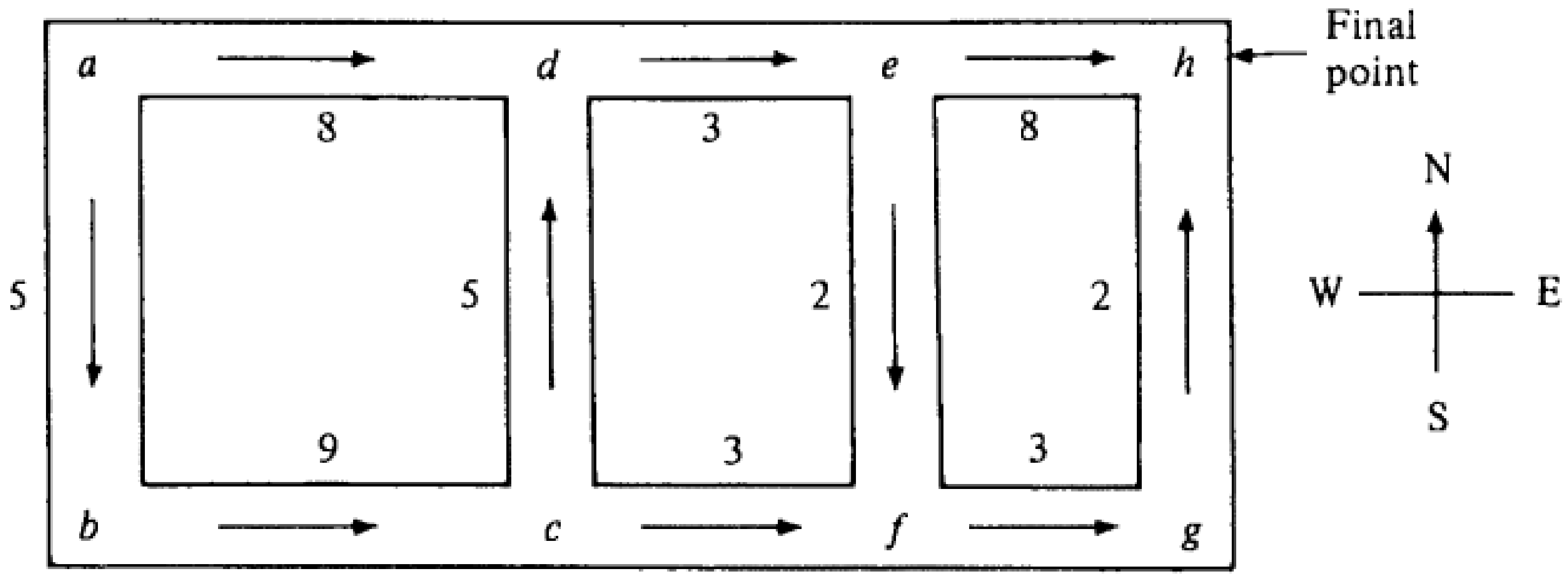
$$C_{bcf} = J_{bc} + J^*_{cf}$$
$$C_{bdf} = J_{bd} + J^*_{df}$$
$$C_{bef} = J_{be} + J^*_{ef}$$



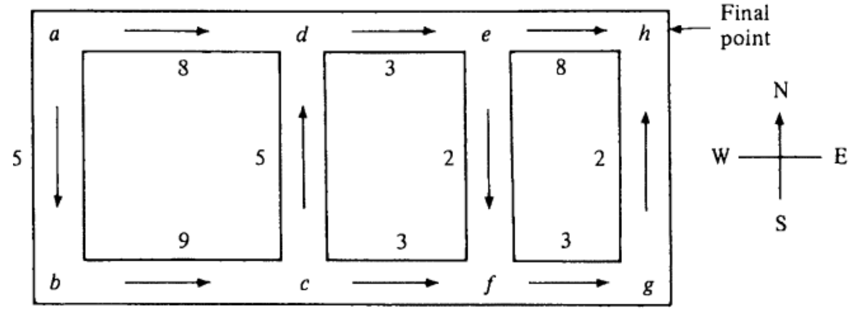
Applying the principle of optimality

- need only to compare the concatenations of immediate decisions and optimal decisions → significant decrease in computation / possibilities
- in practice: carry out this procedure backward in time

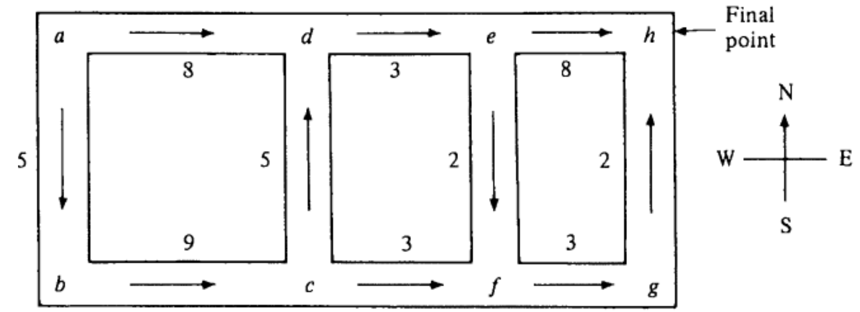
Example



Example



Example



Optimal cost: 18; Optimal path: $a \rightarrow\ d \rightarrow\ e \rightarrow\ f \rightarrow\ g \rightarrow\ h$

DP Algorithm

- Model: $\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, k), \quad \mathbf{u}_k \in U(\mathbf{x}_k)$
- Cost: $J(\mathbf{x}_0) = h_N(\mathbf{x}_N) + \sum_{k=0}^{N-1} g(\mathbf{x}_k, \pi_k(\mathbf{x}_k), k)$

DP Algorithm: For every initial state $\mathbf{x}_0$, the optimal cost $J^*(\mathbf{x}_0)$ is equal to $J_0(\mathbf{x}_0)$, given by the last step of the following algorithm, which proceeds backward in time from stage $N-1$ to stage 0:

$$J_N(\mathbf{x}_N) = h_N(\mathbf{x}_N)$$

$$J_k(\mathbf{x}_k) = \min_{\mathbf{u}_k \in U(\mathbf{x}_k)} g(\mathbf{x}_k, \mathbf{u}_k, k) + J_{k+1}\big(f(\mathbf{x}_k, \mathbf{u}_k, k)\big), \qquad k = 0, \dots, N-1$$

Furthermore, if $\mathbf{u}_k^* = \pi_k^*(\mathbf{x}_k)$ minimizes the right hand side of the above equation for each $\mathbf{x}_k$ and k, the policy $\{\pi_0^*, \pi_1^*, \dots, \pi_{N-1}^*\}$ is optimal

Comments

- discretization (from differential equations to difference equations)
- quantization (from continuous to discrete state variables / controls)
- global minimum
- constraints, in general, simplify the numerical procedure
- optimal control in closed-loop form
- curse of dimensionality

Next time

- Canonical application: Discrete Linear Quadratic Regulator (LQR)
- Stochastic DP

$$V^*(x) = \max_u \left(R(x,u) + \gamma \sum_{x' \in \mathcal{X}} T(x'|x,u)\, V^*(x') \right)$$