Aalto University
School of Electrical
Engineering

ELEC-E8119 Reinforcement Learning Planning in Discrete Space Introduction to Optimal Control

Ville Kyrki

24.9.2019

Today

- Traditional (discrete) planning
- How to plan with complete knowledge?

- Connection to optimal control

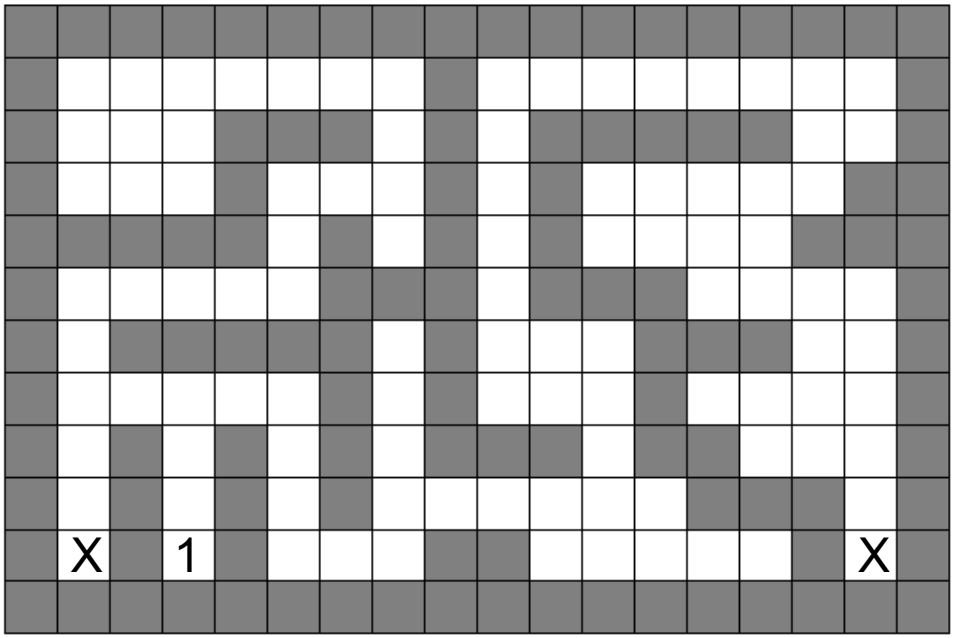
Learning goals

- Formulate discrete planning problems.
- Use value iteration as a way to solve them.

- Explain relationship between discrete planning and optimal control.

Maze

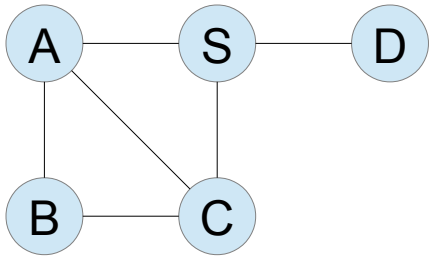
- What are: state space, initial state, goal states set, action space, state transition (dynamics) function?



Planning as graph search – General search algorithm

- Keep track of visited states (dead and alive).
  - Pick an alive state and visit one of the unvisited states (reachable and not dead).
  - Only difference between algorithms how to choose the state to visit.

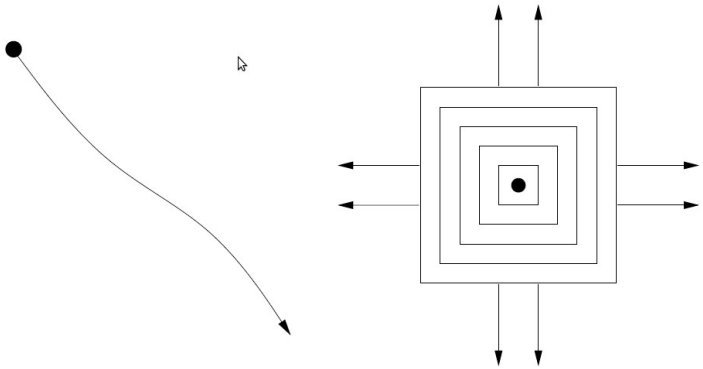
Propose algorithms!



Search strategies

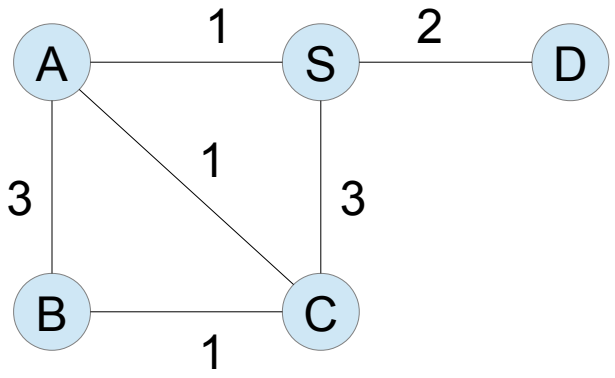
- Are all search strategies good?
  - What properties we want a strategy to have?

- *Systematic* graph search
  - All states will be visited (for finite graphs)
  - If solution exists, it will be found in finite time (for infinite graphs)



Shortest paths

- Associate each edge of graph with nonnegative cost
- Cost of plan is the sum of costs



Shortest path from S to B?

Dijkstra's algorithm

- Idea:
  - Keep track of “cost-to-come”, i.e. accumulated minimum cost to reach a particular state
  - Expand state with minimum cost-to-come
  - Update “cost-to-come” estimates until optimum is reached.
- Computational complexity $O(|V| \log |V| + |E|)$

- Exercise: Solve shortest paths from S in the previous graph using Dijkstra.

- A* is an extension of Dijkstra with heuristic lower bound.

Optimal planning (fixed-length plans)

Like feasible planning, plus

- cost functional

$$L(\tau_K) = \sum_{k=1}^{K} l(x_k, u_k) + l_F(x_{K+1}) \qquad \tau_K = (u_1, \ldots, u_K)$$

$$l_F(x) = \begin{matrix} 0, x \in X_G \\ \infty, x \notin X_G \end{matrix}$$

- Goal: $min_\tau L(\tau)$

Solving optimal planning

- Principle of Optimality (Bellman, 1957): An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

- Value-function:
  - cost-to-go $G_k^*(x_k)$

$$G_k^*(x_k) = min_{u_k, \ldots, u_K} \left\{ \sum_{i=k}^{K} l(x_i, u_i) + l_F(x_F) \right\}$$

Backward value iteration

- Assume we know $G^*_{k+1}(x)$
  how to compute $G^*_k(x_k)$?

$$G^*_k(x_k) = min_{u_k,\ldots,u_K}\left\{\sum_{i=k}^{K} l(x_i, u_i) + l_F(x_F)\right\}$$

$$G^*_k(x_k) = min_{u_k}\left\{l(x_k, u_k) + G^*_{k+1}(x_{k+1})\right\} \qquad G^*_{K+1}(x) = l_F(x)$$

$$= min_{u_k}\left\{l(x_k, u_k) + G^*_{k+1}(f(x_k, u_k))\right\}$$

"Try all actions for this step and choose best."

Value iteration, unknown length plans

- Iterating recursion until value function stationary: optimal cost plans have been found from all states that can reach a goal state

$$G^*(x) = min_u \left\{ l(x,u) + G^*(f(x,u)) \right\}$$

Bellman equation

- Using G^*, optimal actions can be found from

$$u^* = argmin_{u \in U(x)} \left\{ l(x,u) + G^*(f(x,u)) \right\}$$

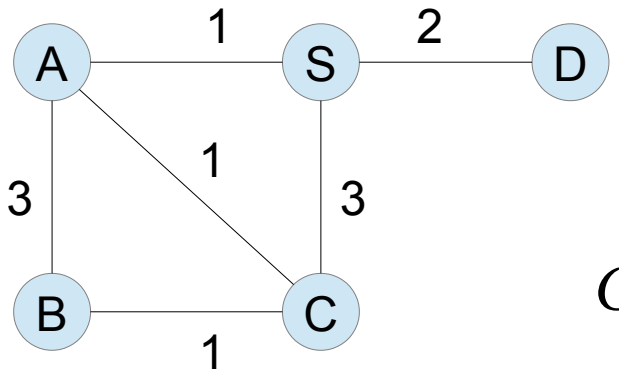
- Complexity $O(K|X||U|) = O(|X|^2|U|)$

Why? Compare to Dijkstra!

Exercise

- Use backward value iteration for

$$x_I = S$$
$$X_G = \{B\}$$



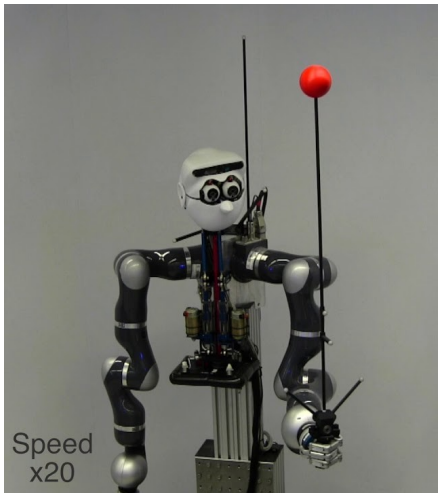
Reminder:

$$G^*(x) = min_u \{ l(x,u) + G^*(f(x,u)) \}$$

What about continuous state variables?

- Simple systems can often be modeled as (locally) linear

$$x_{k+1} = A\,x_k + B\,u_k$$



- Goal is often to reach a particular state with minimum effort.
- Goal can be modeled with a cost function similar to

$$J(u_1, \ldots, u_K) = \sum_{k=1}^{K-1} \left(x_k^T Q\, x_k + u_k^T R\, u_k \right) + x_K^T Q_f\, x_K$$

What are Qs and R?
What is the target state here?

Linear quadratic regulator (LQR)

- Minimize $J(u_1, \ldots, u_K) = \sum_{k=1}^{K-1} \left(x_k^T Q x_k + u_k^T R u_k \right) + x_K^T Q_f x_K$
  for system $x_{k+1} = A x_k + B u_k$
  can be solved by value iteration.

- Value function

$$V_t^*(x_t) = min_{u_t, \ldots, u_K} \sum_{k=t}^{K-1} \left(x_k^T Q x_k + u_k^T R u_k \right) + x_K^T Q_f x_K$$

- Bellman / Hamilton-Jacobi equation

$$V_t^*(x_t) = x_t^T Q x_t + min_{u_t} \left(u_t^T R u_t + V_{t+1}^*(A x_t + B u_t) \right)$$

Solution in the readings for this lecture (there's a minor error, though).

Compare to $G^*(x) = min_u \left\{ l(x, u) + G^*(f(x, u)) \right\}$

Find the error(s)

$$V_t^*(x_t) = x_t^T Q x_t + min_{u_t}\left(u_t^T R u_t + V_{t+1}^*(A x_t + B u_t)\right)$$

$$V_{t+1}^*(x) = x^T P_{t+1} x$$

$$0 = \frac{\partial}{\partial u}\left(u_t^T R u_t + (A x_t + B u_t)^T P_{t+1}(A x_t + B u_t)\right)$$
$$= 2 u^T R + 2 x^T A^T P_{t+1} B + u^T B^T P_{t+1} B$$

Note:

$$\frac{\partial}{\partial x} x^T A x = x^T (A + A^T)$$

Find the error(s)

$$V_t^*(x_t) = x_t^T Q x_t + min_{u_t}\left(u_t^T R u_t + V_{t+1}^*(A x_t + B u_t)\right)$$

$$V_{t+1}^*(x) = x^T P_{t+1} x$$

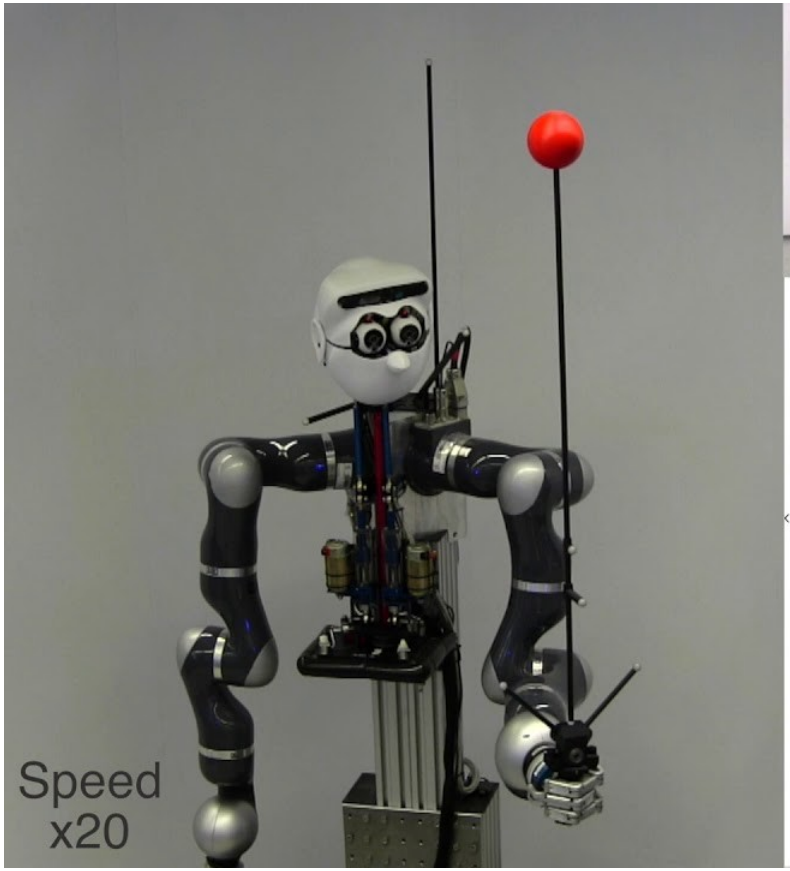
$$0 = \frac{\partial}{\partial u}\left(u_t^T R u_t + (A x_t + B u_t)^T P_{t+1}(A x_t + B u_t)\right)$$
$$= 2 u^T R + 2 x^T A^T P_{t+1} B + u^T B^T P_{t+1} B$$

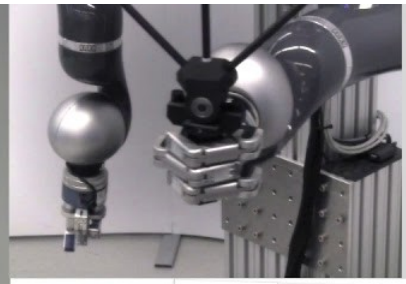
Now solve for u!

Note:

$$\frac{\partial}{\partial x} x^T A x = x^T (A + A^T)$$

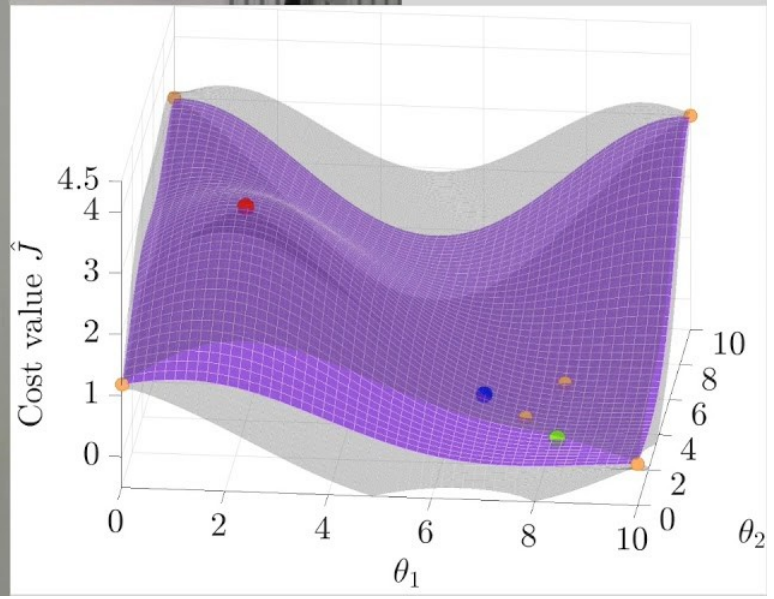
LQR example





Global search

Iteration 3



https://www.youtube.com/watch?v=TrGc4qp3pDM

But what if the dynamics model has uncertainties?

Summary

- Feasible planning can be understood as search.
- Optimal planning can be solved by dynamic programming using a recursive formulation of value function.
- Optimal control of linear systems with quadratic costs can be done using Bellman equation similar to discrete planning.

Next time: Stochastic discrete state environment

- Readings:
  - Sutton & Barto, chapters 2-2.3, 2.5-2.6, 3-3.8 due week from now