24.2.2015

Optimal and Learning Control

for

Autonomous Robots

Lecture 2

A D R L

Jonas Buchli
Agile & Dexterous Robotics Lab

FNSNF

ETH Zürich

Lecture 2 Goals

- ★Discrete optimal control problem
- ★Value function and optimal value function
- ★Bellman Equation
- ★Optimal Bellman Equation
- ★Optimal solution constructed backwards in time (cf. Principle of optimality)

Class logistics

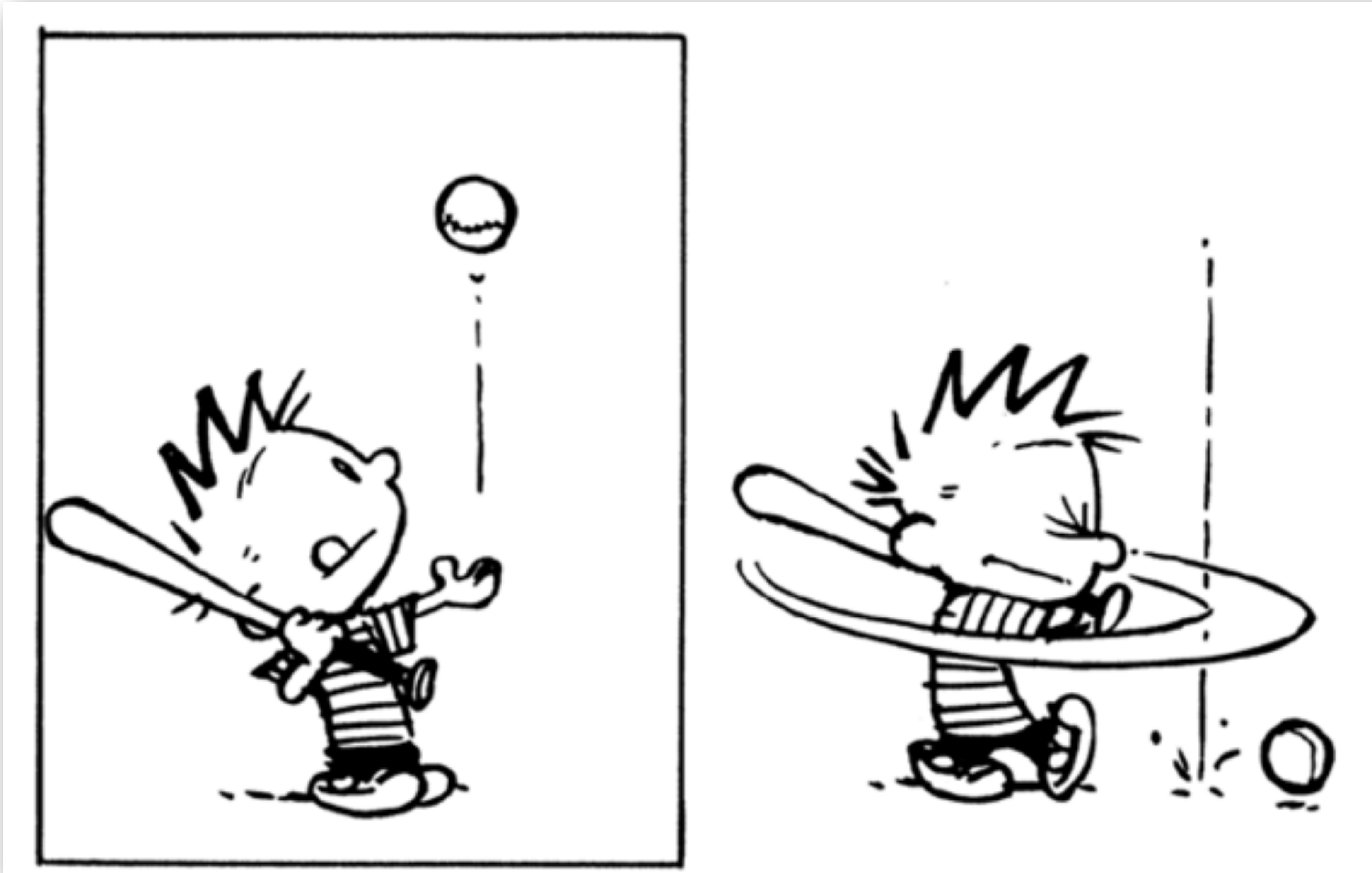
Lecturer: Jonas Buchli - buchlij@ethz.ch
Assistant: Farbod Farshidian - farshidian@mavt.ethz.ch

Office hours: Thu, 18-19 Room: ML J37.1
(no office hours this week, first office hour March 5)

Website:
http://www.adrl.ethz.ch/doku.php/adrl:education:lecture:fs2015

L1 Recap

Reinforcement Learning



Learning from unspecific reward
'by trial and error' - delayed reward

A D R L

ETH Zürich

Cost and reward functions

‘A single number describing the quality of the solution’

Quality dependent on some parameters

☞ Simple example: design a tank, use minimum amount of material

☞ Complicated example: Minimize boarding time of a plane

Analytical optimum

Minima (and maxima) of functions

n-dimensional:

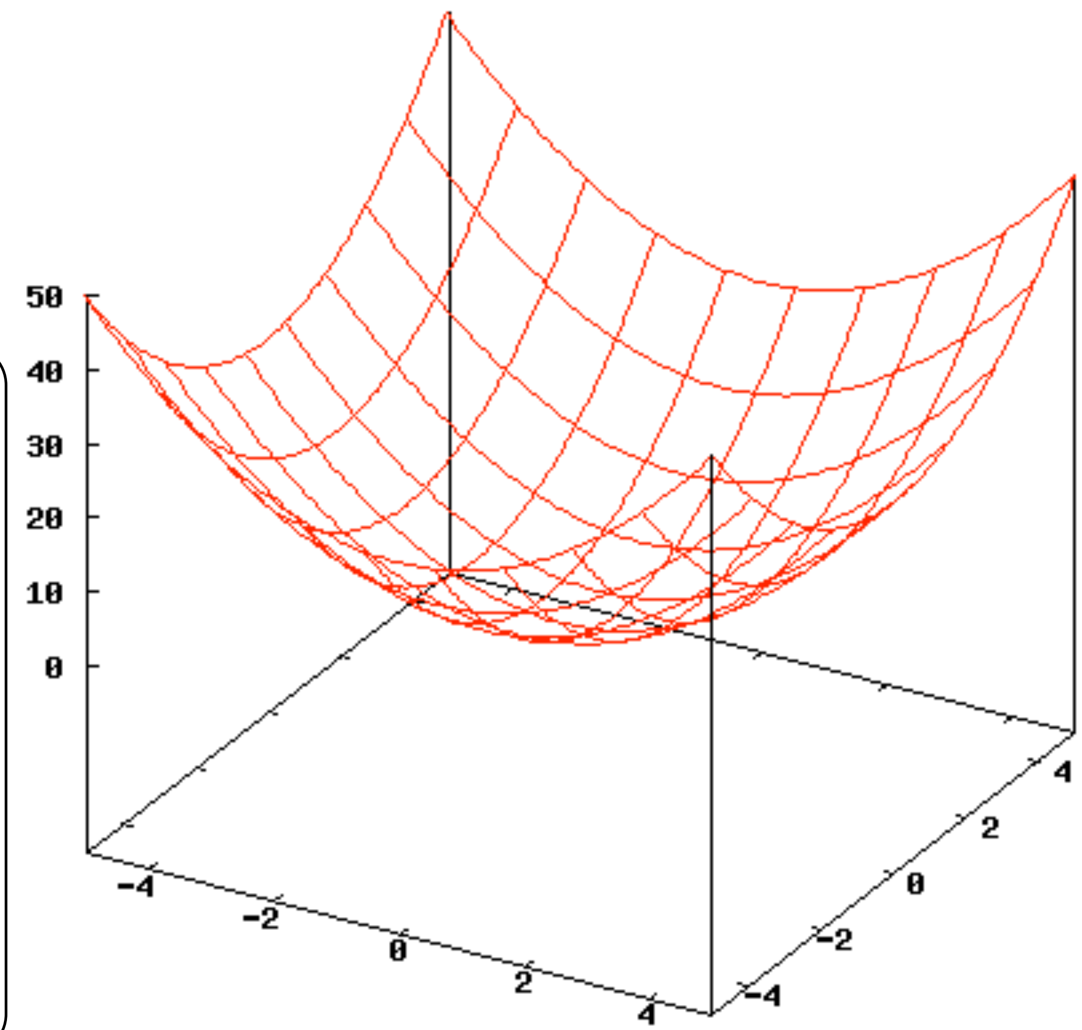
$$C = f(x_1, \ldots, x_n)$$

$$\frac{\partial C}{\partial x_i}$$

$$\frac{\partial C}{\partial x_i} = 0$$

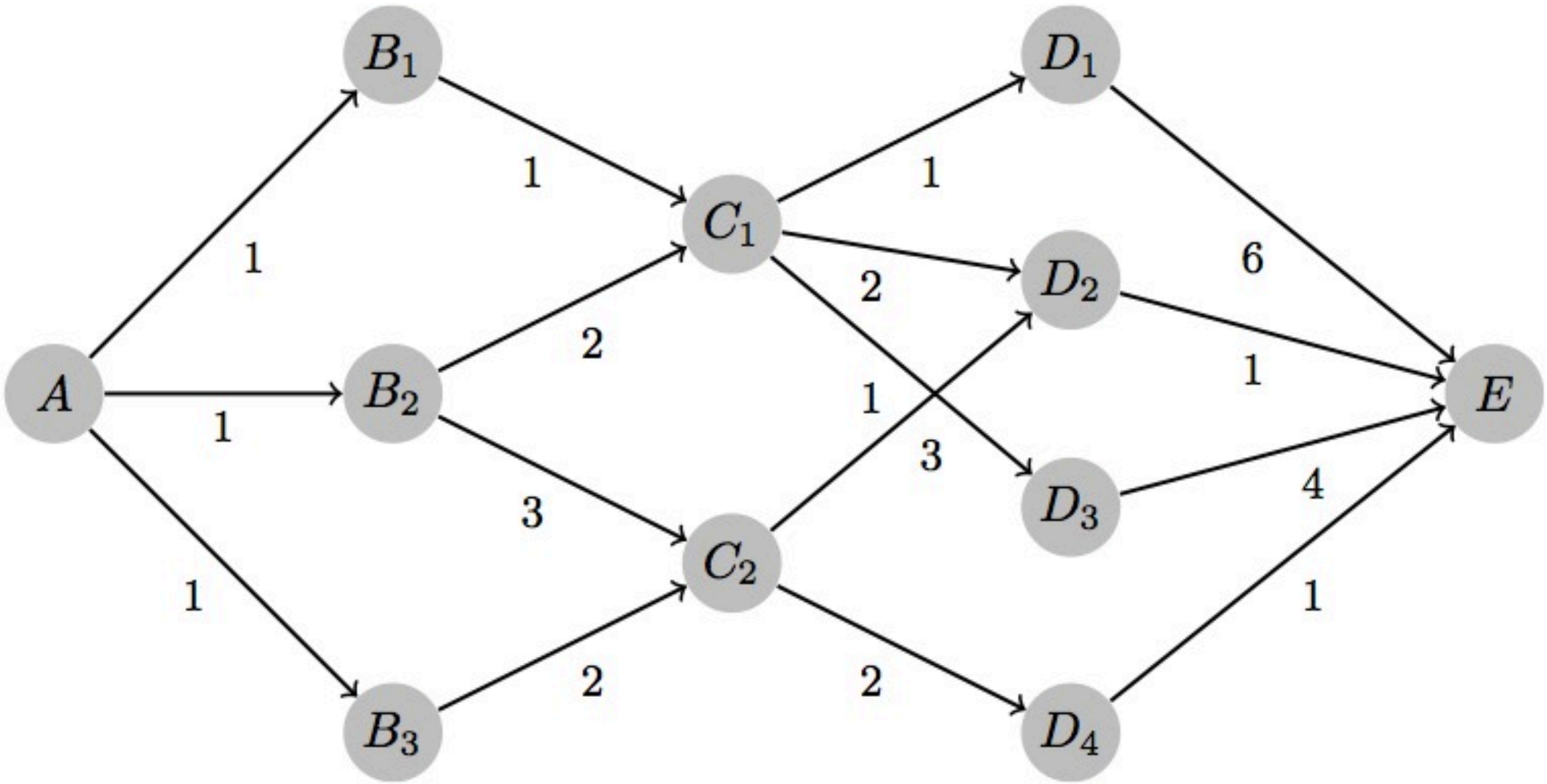
$$\nabla C = \left[\frac{\partial C}{\partial x_1}, \ldots, \frac{\partial C}{\partial x_n}\right]^T$$

$$\nabla C = 0$$



Minimum is an 'inflection point' - slope is 0

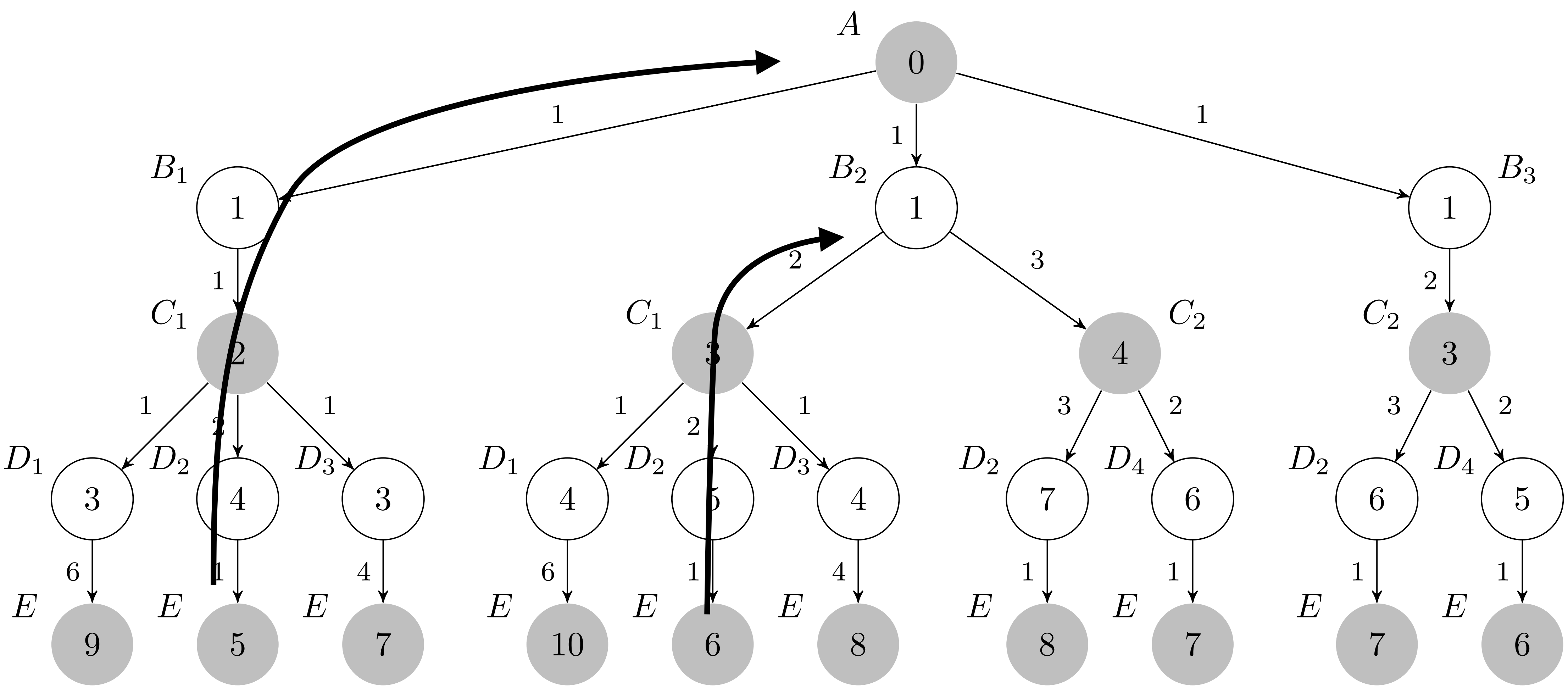
Traveling Salesman



11 nodes
16 edges

Select optimal path?

= control



need to look all the way to the end to find optimal path, local info (edge or next node is not telling)

1.1 Shortest ("cheapest") Path Problem. Example.

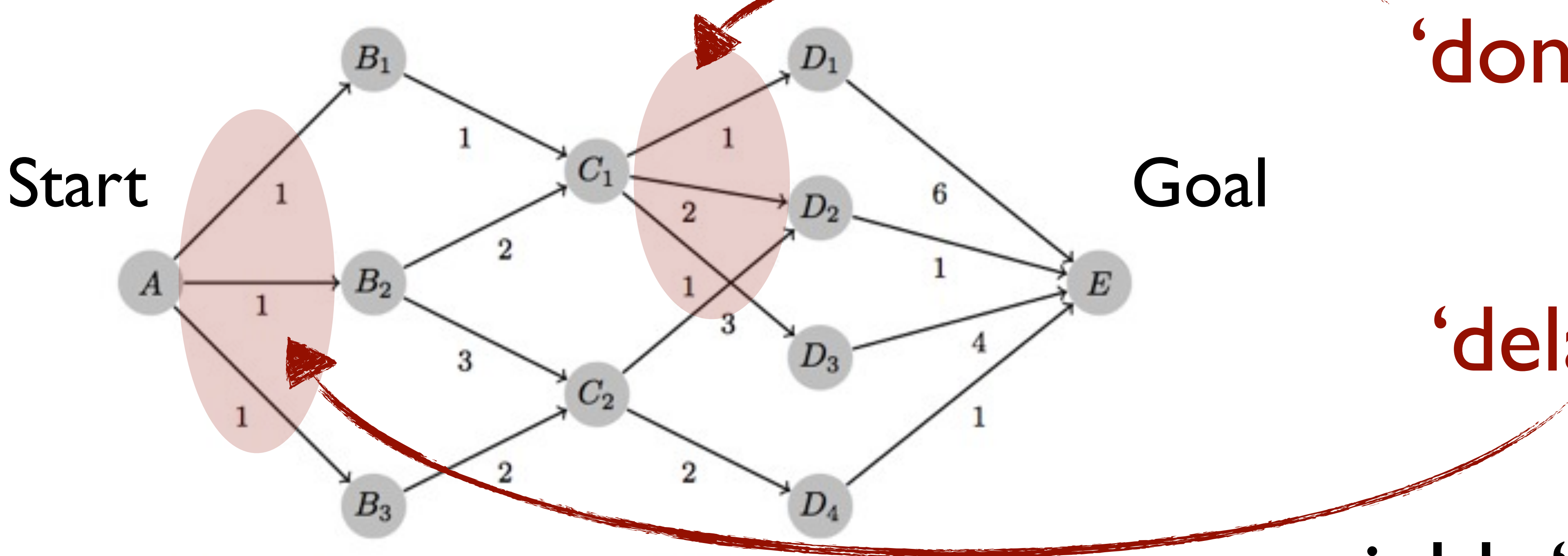


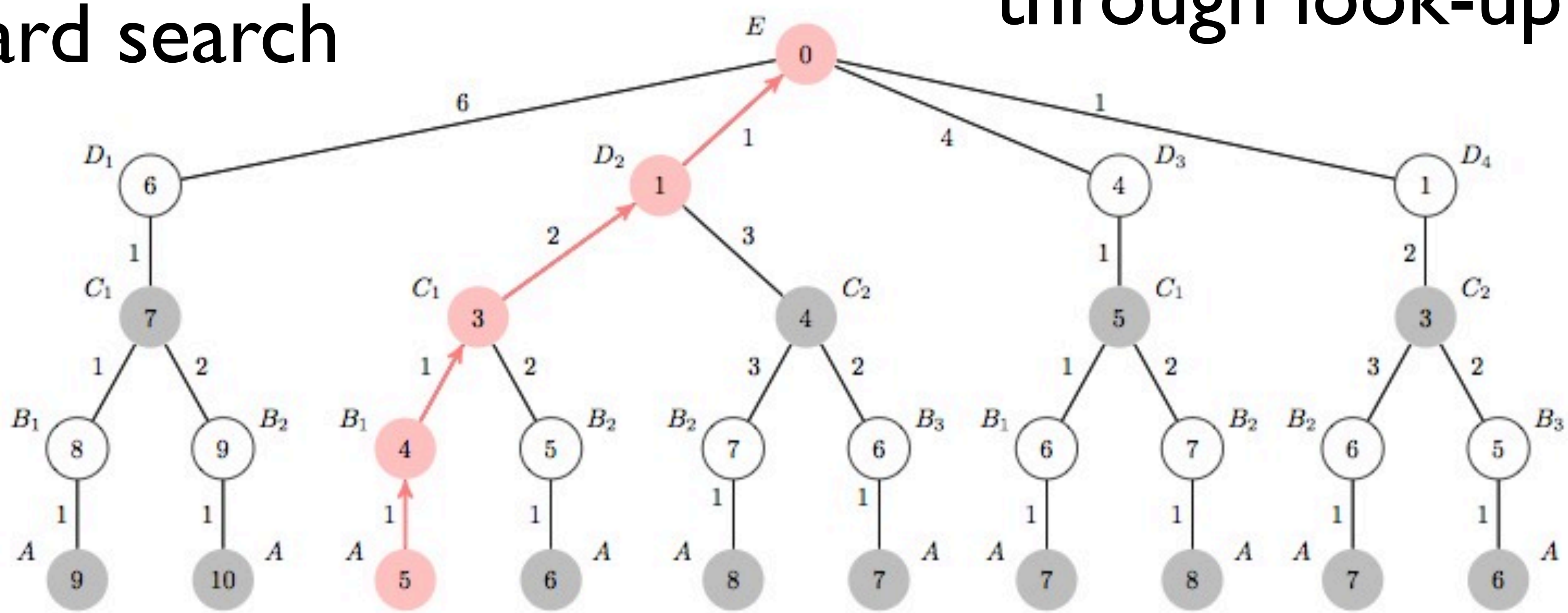
Figure 1: Example of directed graph with cost at each edge.

'don't be greedy!'

'delayed reward'

yields 'control' policy through look-up

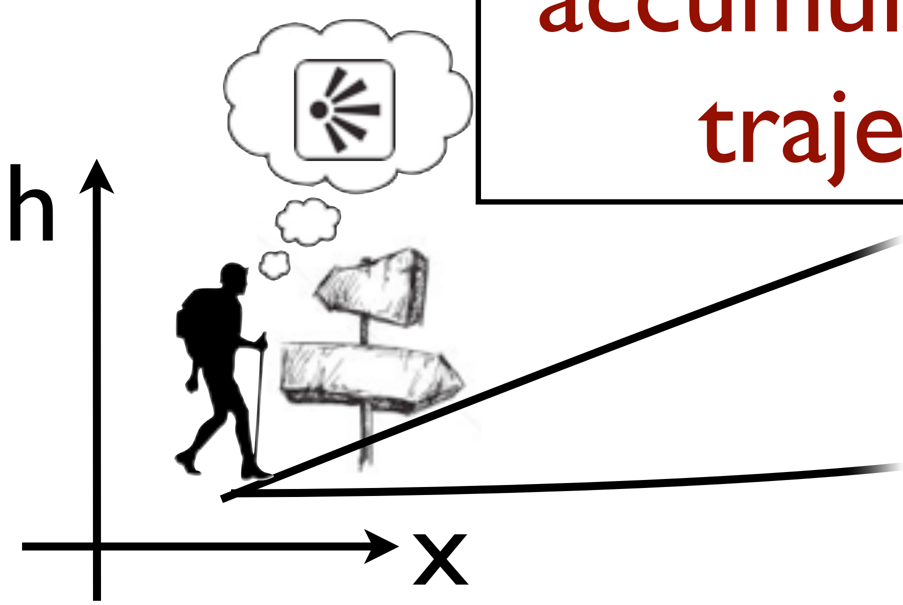
Backward search



Why greedy is not a good idea...

$$R(\boldsymbol{\tau}_i) = \phi_{t_N} + \int_{t_i}^{t_N} r_t \, dt$$

accumulated reward is a function of a trajectory through state space



value function is a function of the state!

THM: Need to look all the way to the end to know what's optimal!

$$V = \text{'total expected height gain from this position'}$$

Value function!

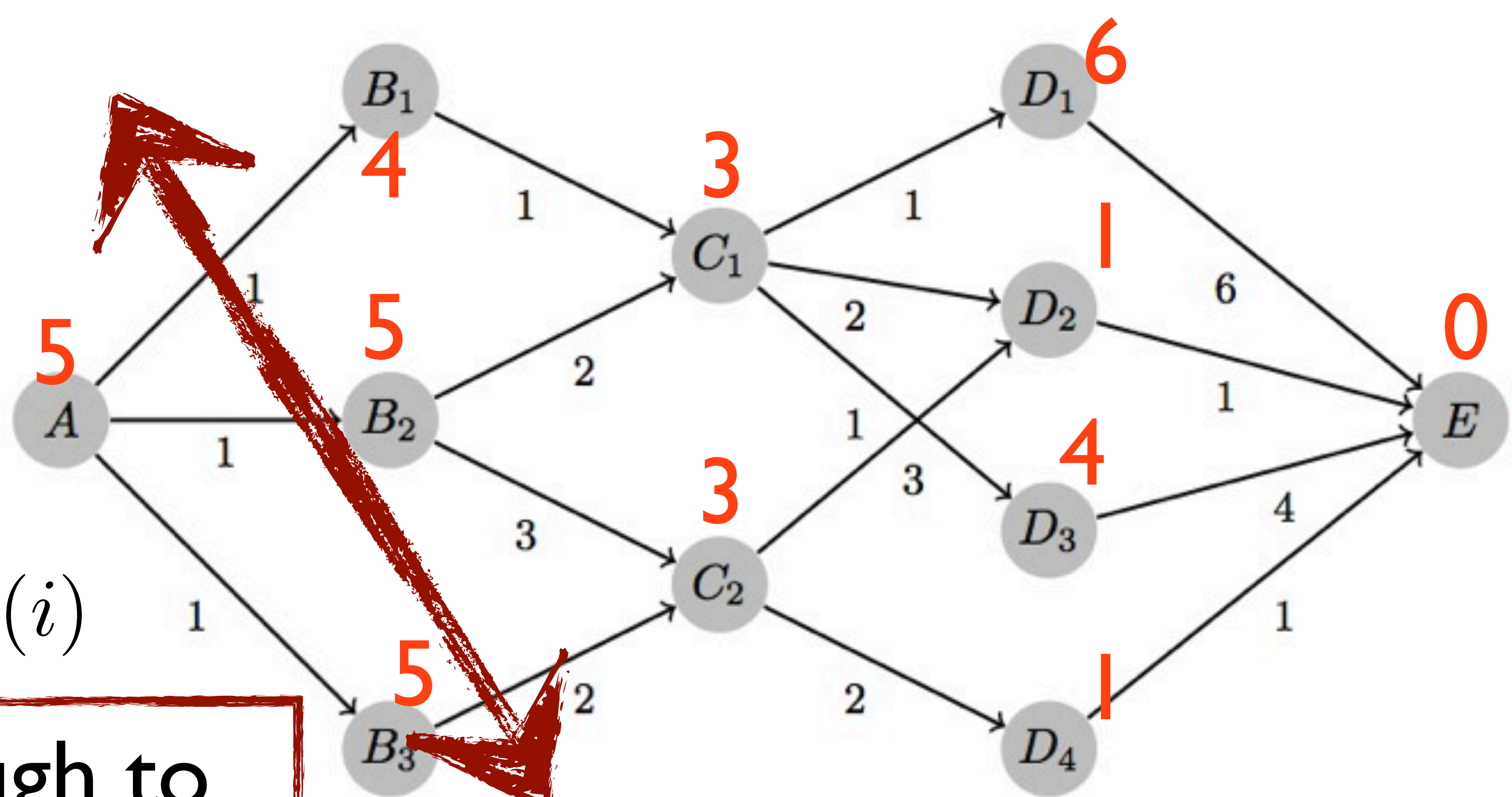
Map of shortest achievable distance

'How long is the shortest path from here if following the optimal route?'

'How valuable is a position?'
'What cost can I expect?'

$$V(n) = \sum_{i=N}^{n} e(i)$$

$$V(n) = e(n) + \sum_{i=N}^{n+1} e(i)$$



'local info' is enough to determine next step!!!

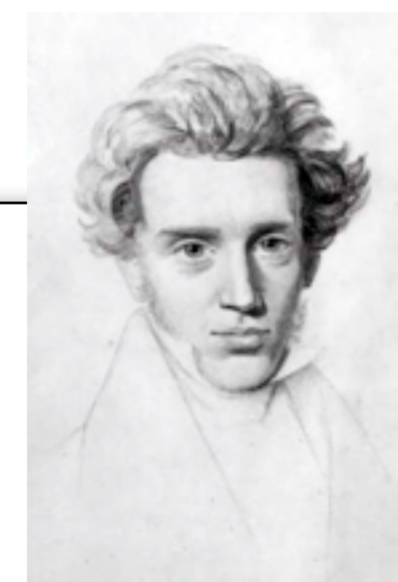
Value function

Principle of optimality

If a path ABCDE is optimal, then all parts of this path starting at intermediate position and ending at E (BCDE,CDE,DE) are optimal.

Life can only be understood backwards; but it must be lived forwards.

Søren Kierkegaard (1813-55)

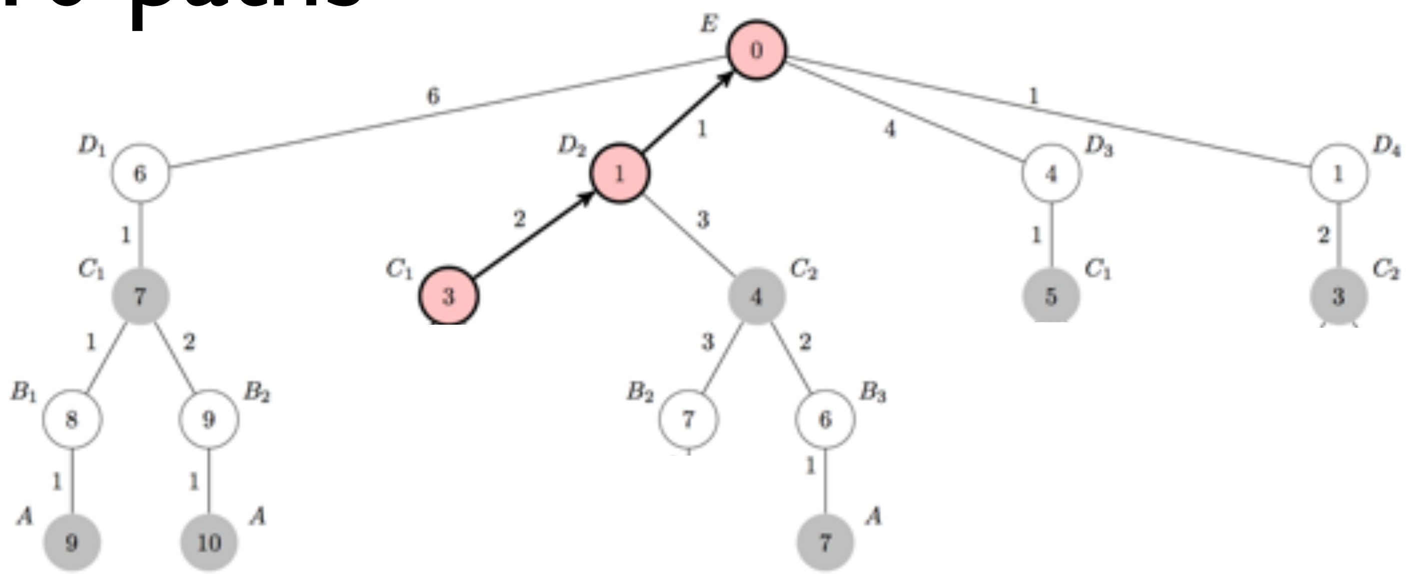


EOF Recap

Computational complexity

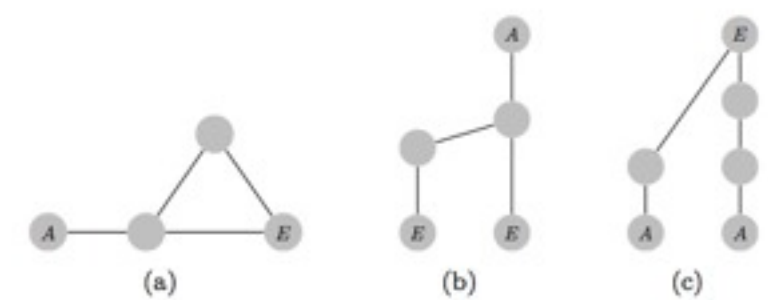
Original graph: 11 nodes -16 edges

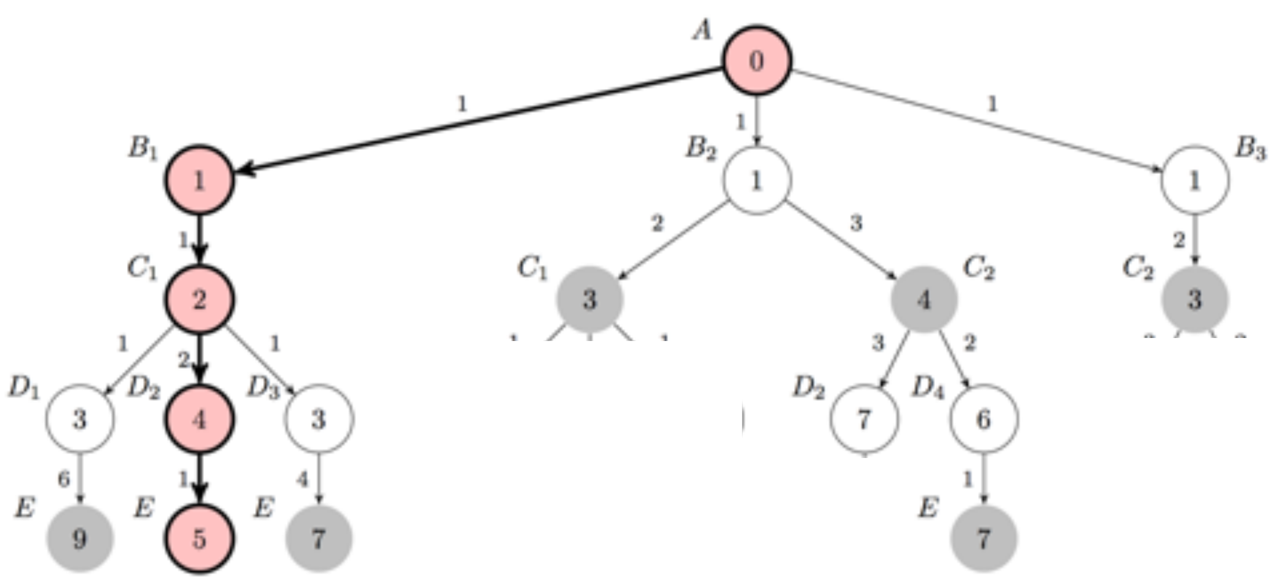
10 paths



30 nodes
29 edges

pruning ⇒17N/16E





28 nodes
27 edges

pruning ⇒ 17N/16E

A D R L

ETH zürich

(Memory) complexity of decision trees

Size of regular tree
w. branching factor b, depth n: b^n

Examples for branching factors:

Rubik cube ~13.34
Chess ~35
Go 250

12 DOF Robot?
Naive assumptions:
Resolution of motor commands 1/1000: b=1000^12

BUT Physics is (mostly) ‘smooth’: ... Similar nodes, similar subtree

Taxonomy of dynamic systems

- ★ time: continuous vs discrete
- ★ state: continuous vs. discrete
- ★ linear vs. nonlinear
- ★ deterministic vs. stochastic

time

states	discrete	continuous
discrete	gridworlds Markov chains	Petrinets Discrete Event Systems
continuous	$x_{n+1} = f(x_n, n)$ Map Difference equations $\Delta x_{n+1} = f(x_n, n)$	Differential equations

L1 example - TSP

'Robots'
Newtonian Physics

Discrete finite time, deterministic system & cost function

discrete time

$$x_n = x(t_n)$$

$$n \in \{0, 1, ..., N-1\}$$

Given system with dynamics

$$x_{n+1} = f_n(x_n, u_n) \qquad \text{given } x(0) = x_0$$

and cost

$$J = \alpha^N \Phi(x_N) + \sum_{k=0}^{N-1} \alpha^k L_k(x_k, u_k)$$

$$0 \leq \alpha \leq 1$$

discount/decay factor

n is the discrete time index,

x_n is the state of the system at time n,

u_n is the control input at time n and

f_n is the the state transition equation.

Discrete optimal control problem

finite time, deterministic

Find control $u_k^* = \mu^*(k, x_k)$ minimizing

$$J = \alpha^N \Phi(x_N) + \sum_{k=0}^{N-1} \alpha^k L_k(x_k, u_k)$$

Given constraints

$$x_{n+1} = f_n(x_n, u_n)$$

Goal: Optimal policy

$$\mu^* = \arg\min_u J$$

Value function

Value function for policy μ

$$V^{\mu}(n,x) = \alpha^{N-n}\Phi(x_N) + \sum_{k=n}^{N-1} \alpha^{k-n} L_k(x_k,u_k)$$

Generally value function time and state dependent!

$$x_n = x$$

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

$$u_k = \mu(k,x_k)$$

Value function for final time equals cost at final time

$$V^{\mu}(N,x) = \Phi(x)$$

Cost vs. Value function

$$J = \alpha^N \Phi(x_N) + \sum_{k=0}^{N-1} \alpha^k L_k(x_k, u_k)$$

$$V^\mu(n, x) = \alpha^{N-n} \Phi(x_N) + \sum_{k=n}^{N-1} \alpha^{k-n} L_k(x_k, u_k)$$

Effect of final cost becomes more prominent

Cost equals Value function at time 0

$$J = V(0, x_0)$$

A D R L

ETH Zürich

Bellman equation

Derivation

Starting with Value function

$$V^\mu(n,x) = \alpha^{N-n}\Phi(x_N) + \sum_{k=n}^{N-1} \alpha^{k-n} L_k(x_k,u_k)$$

factoring out first step

$$V^\mu(n,x) = L_n(x,u_n) + \alpha^{N-n}\Phi(x_N) + \sum_{k=n+1}^{N-1} \alpha^{k-n} L_k(x_k,u_k)$$

$$V^\mu(n,x) = L_n(x,u_n) + \alpha\left[\alpha^{N-n-1}\Phi(x_N) + \sum_{k=n+1}^{N-1} \alpha^{k-n-1} L_k(x_k,u_k)\right]$$

compare $V^\mu(n+1, x_{n+1})$

$$V^\mu(n+1, x_{n+1}) \qquad x_{n+1} = f(x,u_n)$$

Bellman equation

$$V^\mu(n,\mathbf{x}) = L_n(\mathbf{x},\mathbf{u}_n) + \alpha V^\mu\left(n+1, f_n\left(\mathbf{x},\mathbf{u}_n\right)\right)$$

final condition $V^\mu(N,x) = \Phi(x)$

The backwards nature of the value function

Bellman equation

$$V^{\mu}(n+1, x_{n+1})$$

$$V^{\mu}(n, \mathbf{x}) = L_n(\mathbf{x}, \mathbf{u}_n) + \alpha V^{\mu}\left(n+1, f_n\left(\mathbf{x}, \mathbf{u}_n\right)\right)$$

final condition $V^{\mu}(N, x) = \Phi(x)$

If I want to know V at given node n, need to start with final value and compute backwards

A D R L

ETH Zürich

Optimal policy

optimal value function

$$V^*(n,x) \leq V^\mu(n,x) \qquad \forall n, x$$

Remember:V is based on cost ⇒ minimize

equivalent notation

$$V^*(n,x) = \min_\mu V^\mu(n,x) \qquad \forall n, x$$

Optimal policy is the one that minimizes RHS

$$\mu^* = \{\mathbf{u}_n^*, \ldots, \mathbf{u}_{N-1}^*\} = \arg\min_\mu V^\mu(n,\mathbf{x}) \qquad \forall n : 0, \ldots, N-1$$

substitute Bellman Equation into V^μ

$$V^\mu(n,\mathbf{x}) = L_n(\mathbf{x},\mathbf{u}_n) + \alpha V^\mu\left(n+1, f_n\left(\mathbf{x},\mathbf{u}_n\right)\right)$$

$$V^*(n,\mathbf{x}) = \min_{\mathbf{u}_n}\left[L_n(\mathbf{x},\mathbf{u}_n) + \alpha V^*\left(n+1, \mathbf{f}_n\left(\mathbf{x},\mathbf{u}_n\right)\right)\right]$$

Optimal Bellman Equation

A D R L

ETH Zürich

Optimal Bellman Equation

$$V^*(n,\mathbf{x}) = \min_{\mathbf{u}_n} \left[L_n(\mathbf{x},\mathbf{u}_n) + \alpha V^*\left(n+1, \mathbf{f}_n\left(\mathbf{x},\mathbf{u}_n\right)\right)\right]$$

★ Optimal Bellman Eq. computes
Optimal Value function

if u continuous:

$$\frac{\partial}{\partial \mathbf{u}_n}\left[L_n(\mathbf{x},\mathbf{u}_n) + \alpha V^*\left(n, \mathbf{f}_n\left(\mathbf{x},\mathbf{u}_n\right)\right)\right] = 0$$

- Bellman Equation requires working 'backwards in time' / from end to start
- Bellman Equation allows to find optimal solution one step at a time
- ... whereas Value function requires optimization of the whole control sequence at once

$$V^{\mu}(n,x) = \alpha^{N-n}\Phi(x_N) + \sum_{k=n}^{N-1} \alpha^{k-n} L_k(x_k,u_k)$$

ADRL

ETH Zürich

Optimal Control

$$\mathbf{u}^*(n, \mathbf{x}) = \arg\min_{\mathbf{u}_n} \left[L_n(\mathbf{x}, \mathbf{u}_n) + \alpha V^* \left(n+1, \mathbf{f}_n\left(\mathbf{x}, \mathbf{u}_n\right)\right)\right]$$

A D R L

Optimal value and control

$$\mathbf{u}^*(n, \mathbf{x}) = \arg\min_{\mathbf{u}_n} \left[L_n(\mathbf{x}, \mathbf{u}_n) + \alpha V^*\left(n+1, \mathbf{f}_n\left(\mathbf{x}, \mathbf{u}_n\right)\right)\right]$$

final condition $V^*(N, x) = \Phi(x)$

```
(1)    init:
       set n=N
       compute final cost set V(N) = final cost
```

$$V^*(n-1, \mathbf{x}) = \min_{\mathbf{u}_{n-1}} \left[L_{n-1}(\mathbf{x}, \mathbf{u}_{n-1}) + \alpha V^*\left(n, \mathbf{f}_{n-1}\left(\mathbf{x}, \mathbf{u}_{n-1}\right)\right)\right]$$

```
(2)    'for all' x_{n} compute Value function: V*(n-1,x)
       -> optimal control at step n-1, u_opt(x,n-1)
```

$$\mathbf{u}^*(n-1, \mathbf{x}) = \arg\min_{\mathbf{u}_{n-1}} \left[L_{n-1}(\mathbf{x}, \mathbf{u}_{n-1}) + \alpha V^*\left(n, \mathbf{f}_{n-1}\left(\mathbf{x}, \mathbf{u}_{n-1}\right)\right)\right]$$

```
(4)    n = n-1
(5)    if (n == 0) : halt, else: goto step 2
```

$$\frac{\partial}{\partial \mathbf{u}_{n-1}} \left[L_{n-1}(\mathbf{x}, \mathbf{u}_{n-1}) + \alpha V^*\left(n, \mathbf{f}_{n-1}\left(\mathbf{x}, \mathbf{u}_{n-1}\right)\right)\right] = 0$$

numerical root finding - iterative method

Optimal control along optimal trajectory

- Note the cost of evaluation (at each time step, for each state, 'try' all controls)

➡ Instead of whole state space: neighborhood of optimal solution
➡ Chicken & egg: What's the optimal solution
➡ Initial guess
➡ Requires other type of algorithms (e.g. ILQC)
➡ Approximations

ADRL

ETH Zürich

Important note: picking an initial x(0) uniquely determines the optimal sequence both in state and controls

Infinite time horizon

choose such α that cost is finite

$$J = \sum_{k=0}^{\infty} \alpha^k L(x_k, u_k), \qquad \alpha \in [0, 1]$$

$$V^*(n, x) = \min_{\mu} \left[\sum_{k=n}^{\infty} \alpha^{k-n} L(x_k, u_k) \right]$$

$$V^*(n + \Delta n, x) = \min_{\mu} \left[\sum_{k=n+\Delta n}^{\infty} \alpha^{k-n-\Delta n} L(x_k, u_k) \right]$$

$$k' = k - \Delta n$$

$$k = k' + \Delta n$$

$$= \min_{\mu} \left[\sum_{k'=n}^{\infty} \alpha^{k'-n} L(x_{k'+\Delta n}, u_{k'+\Delta n}) \right]$$

independent of time → $x_{k+1} = f(x_k, u_k)$

$$x(n + \Delta n) = x(n)$$

$$\boxed{V^*(n, x) = V^*(n + \Delta n, x) = V^*(x)}$$

A D R L

ETH Zürich

Finite vs. infinite

If a final value (finite time), time matters, i.e. it matters at what time in given state

If no final value (infinite time) only state matters

Bellman Equation

discrete, deterministic, infinite time

$$V^*(n, x) = V^*(n + \Delta n, x) = V^*(x)$$

$$V^*(x) = \min_u \{L(x, u) + \alpha V^*(f(x, u))\}$$

Stochastic system

$$x_{n+1} = f(x_n, u_n) + w_n$$

Additive noise

$$w_n \sim P_w(\cdot \mid x_n, u_n)$$

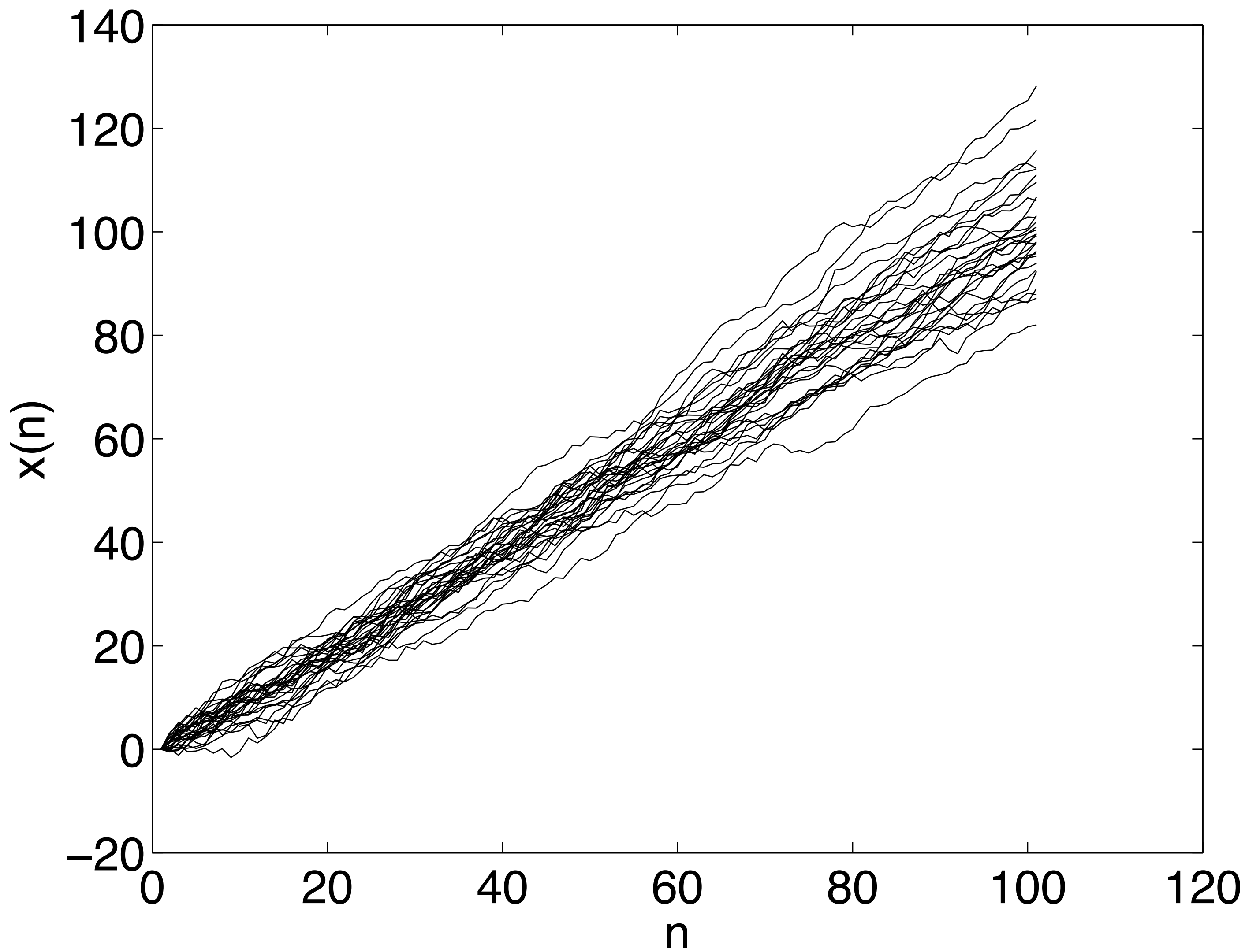
Conditional Probability Distribution
'function of state and control'

$$x_{n+1} = x'$$

General stochastic dynamics

$$x' \sim P_f(\cdot \mid x_n, u_n)$$

Example



$$x(n+1) = x(n) + c + w$$

$$w \sim N(0, \sigma)$$

Cost in stochastic system?

★ Even if we keep u fixed, path x(0...N) will be different each time

➡ thus so is cost

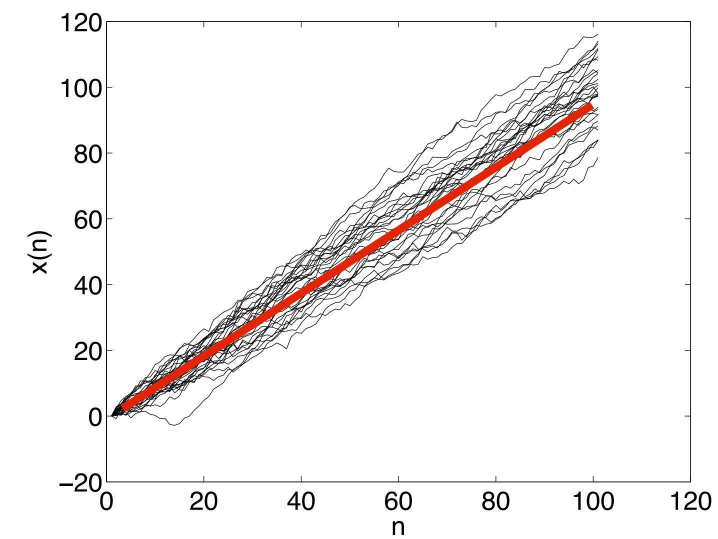
So how to minimize the cost???

✳ Idea: minimize 'in average', i.e. find best solution in average

- average = expected value

➡ minimize expected cost

Expectation



Expected value of x:

Discrete states

$$E(x) = \sum_i P(x_i) x_i \approx \sum_s \frac{1}{N} x_s$$

$$x_s \sim P(x)$$

$$\sum_i P(x_i) = 1$$

$$P(x) \geq 0$$

'weighted average'

Continuous states

$$E(x) = \int p(x) x dx \approx \sum_s \frac{1}{N} x_s$$

$$\int p(x) dx = 1$$

Mathematical expectation itself is not a random variable!
Numerical approximation is a random variable.

A D R L

ETH Zürich

Conditional probability and expectation

$$E(x|y) = \sum_i P(x_i, y)x_i$$

$$E(x|y) = \int p(x, y)xdx$$

Cost in stochastic problem

Expected cost:

$$J = E\left[\alpha^N \Phi(x_N) + \sum_{k=0}^{N-1} \alpha^k L_k(x_k, u_k)\right]$$

Cost is weighted average of all possible costs
Weight = probability of outcome

In stochastic optimal control: Can not optimize outcome, but only the average outcome (expected outcome). The actual cost in a 'rollout' will always be different from the expected cost.

Value functions

Value function for policy

$$V^{\mu}(n,x) = E\left[\alpha^{N-n}\Phi(x_N) + \sum_{k=n}^{N-1}\alpha^{k-n}L_k(x_k,u_k)\right]$$

Optimal value function

$$V^{*}(n,x) = \min_{\mu} E\left[\alpha^{N}\Phi(x_N) + \sum_{k=n}^{N-1}\alpha^{k-n}L_k(x_k,u_k)\right]$$

Optimal policy

$$\mu^{*} = \arg\min_{\mu} E\left[\alpha^{N-n}\Phi(x_N) + \sum_{k=n}^{N-1}\alpha^{k-n}L_k(x_k,u_k)\right]$$

Value function and optimal policy are deterministic (but a function of probability distribution P)

A D R L

ETH Zürich

Bellman equation

$$E(x) = \int p(x) x dx$$

sum over all x'

$$V^{\mu}(n,x) = L_n(x,u_n) + E_{x' \sim P_f(.|x,u_n)}\left[V^{\mu}\left(n+1,x'\right)\right]$$

Optimal Bellman Equation

$$V^{*}(n,x) = \min_{u_n}\left[L_n(x,u_n) + E_{x' \sim P_f(\cdot|x,u_n)}\left[V^{*}\left(n+1,x'\right)\right]\right]$$

Optimal Control

$$u^{*}(n) = \arg\min_{u_n}\left[L_n(x,u_n) + E_{x' \sim P_f(\cdot|x,u_n)}\left[V^{*}\left(n+1,f_n\left(x,u_n\right)\right)\right]\right]$$

x' conditioned on x(n) and u(n)

optimal control is deterministic, not a random variable!

A D R L

ETH Zürich

Optimal control

$$u^*(n) = \arg\min_{u_n} \left[L_n(x, u_n) + E_{x' \sim P_f(\cdot|x,u_n)} \left[V^*\left(n+1, f_n\left(x, u_n\right)\right) \right] \right]$$

final condition $V^*(N, x) = \Phi(x)$

```
(1)   init:
      compute final cost
      set n=N, set V(N) = final cost
```

(deterministic

$$V^*(n-1, \mathbf{x}) = \min_{\mathbf{u}_{n-1}} \left[L_{n-1}(\mathbf{x}, \mathbf{u}_{n-1}) + \alpha V^*\left(n, \mathbf{f}_{n-1}\left(\mathbf{x}, \mathbf{u}_{n-1}\right)\right) \right]$$

)

sum over all x'

$$V^*(n-1, \mathbf{x}) = \min_{\mathbf{u}_{n-1}} \left[L_{n-1}(\mathbf{x}, \mathbf{u}_{n-1}) + E_{\mathbf{x}' \sim P_f(\cdot|\mathbf{x},\mathbf{u}_{n-1})} \left[V^*\left(n, \mathbf{x}'\right) \right] \right]$$

```
(2)   'for all' x_{n} compute Value function: V*(n-1,x)
      -> optimal control at step n-1, u_opt(x,n-1)

(4)   n = n-1
(5)   if (n == 0) : halt, else: goto step 2
```

'sum'

$$\mathbf{u}^*(n-1, \mathbf{x}) = \arg\min_{\mathbf{u}_{n-1}} \left[L_{n-1}(\mathbf{x}, \mathbf{u}_{n-1}) + E_{\mathbf{x}' \sim P_f(\cdot|\mathbf{x},\mathbf{u}_{n-1})} \left[V^*\left(n, \mathbf{x}'\right) \right] \right]$$

A D R L

ETH Zürich

Computational complexity of stochastic problem

- Note the cost of naive evaluation is even higher for stochastic systems (at each time step, for each state, ‘try’ all controls, with all possible ‘random’ events)

Infinite, stochastic

$$\mathbf{x}_{n+1} = \mathbf{f}(\mathbf{x}_n, \mathbf{u}_n) + \mathbf{w}_n$$

Note: no time dependency

$$J = E\left[\sum_{k=0}^{\infty} \alpha^k L(\mathbf{x}_k, \mathbf{u}_k)\right] \qquad \alpha \in [0, 1)$$

A D R L

ETH Zürich

Opt. value and Bellman Eq.

$$V^*(\mathbf{x}) = \min_{\mu} E\left[\sum_{n=0}^{\infty} \alpha^n L(\mathbf{x}_n, \mathbf{u}_n)\right]$$

$$V^*(\mathbf{x}) = \min_{\mathbf{u}_n}\{L(\mathbf{x}, \mathbf{u}) + \alpha E[V^*(\mathbf{x}')]\}$$