ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

OLCAR Lecture #2 - 24.9.13

Outline

(1) Principle of optimality
- Shortest path problem
- => Forward solution
- => Backward solution

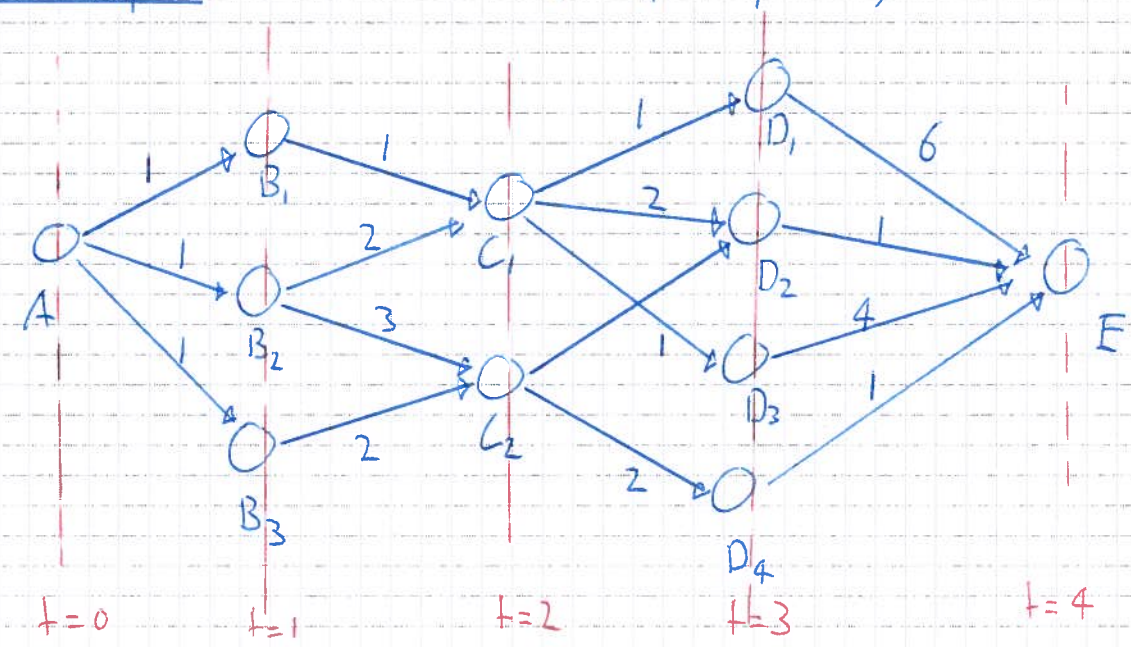
(2) Bellman Equation
- finite vs. infinite

(3) Hamilton-Jacobi-Bellman Equation

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

OLCAR Lecture #2 24.9.13

Example: Shortest (= „cheapest") Path Problem



Directed graph w. cost at each edge

Goal: Find cheapest / shortest Path

① Vanilla / Naive method

Start at A, compute all costs of all paths

Forward search ⟹ Decision tree starting at A

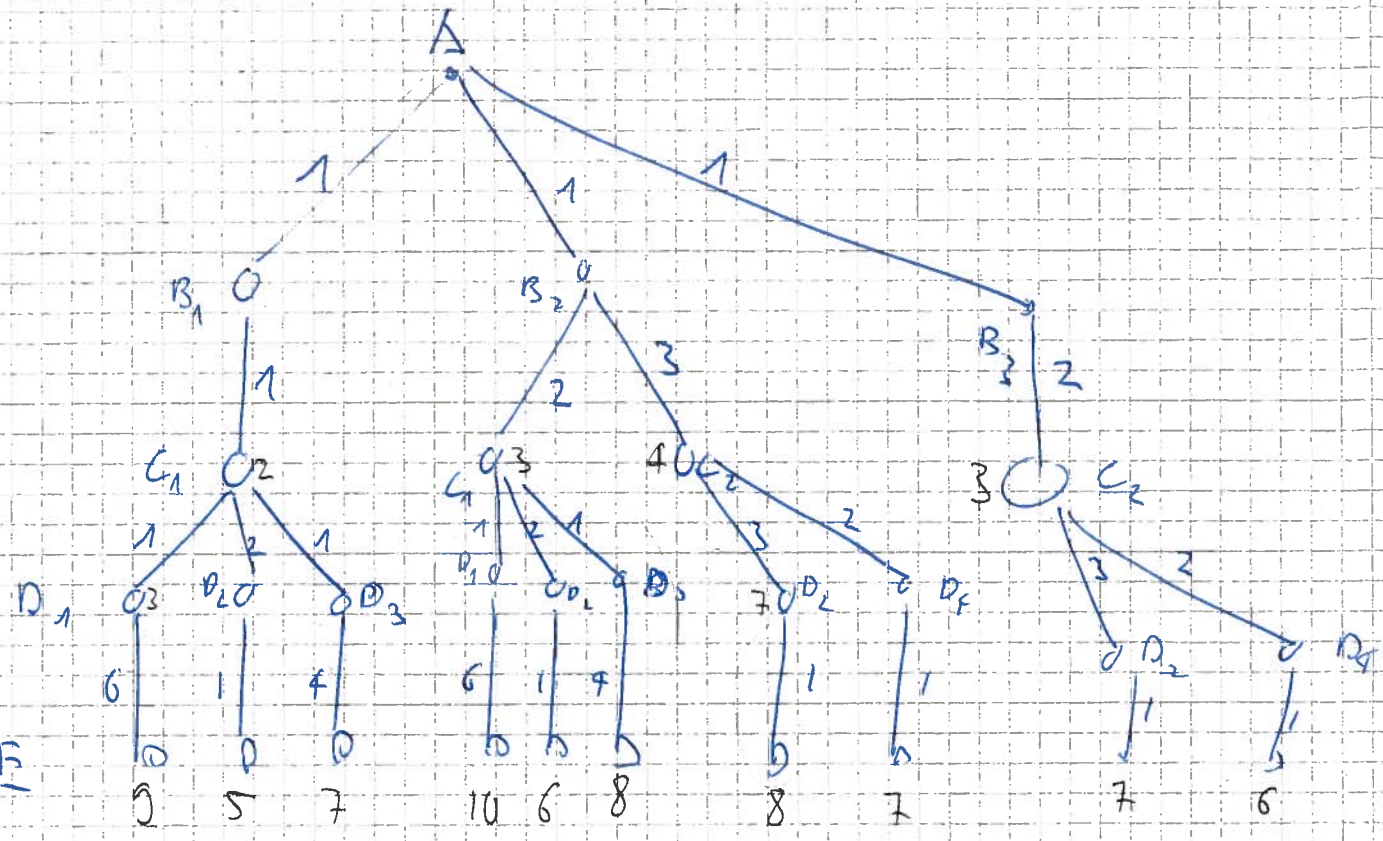
⇒ Lots of unnecessary computation

⇒ Idea „remember" and prune subtrees that have already been evaluated

⇒ Information gets back propagated in „time"

New idea: Backwards search

Forward Search



⇒ Shortest A B_1 C_1 D_2 E

A few observations:

⇒ Sometimes need to take non-greedy decisions
C_1 D_2 has higher cost than alternatives

⇒ Previously: 27 Edges, 27 Evaluations (graph: 10 Paths, 16 Edges)

⇒ Info at nodes useful for decisions
"local controller"? That can not look into future?
⇒ NO

⇒ Observation: Some subtrees are the same!
⇒ Could reuse the info
⇒ Would "back propagate" or back up info
(backwards in time!)
⇒ 16 evaluations ⇔ # Edges?

⇒ What we did here is forward solution!

OLCAR Lecture #2 24.9.13

(Shortest path example cont'd)

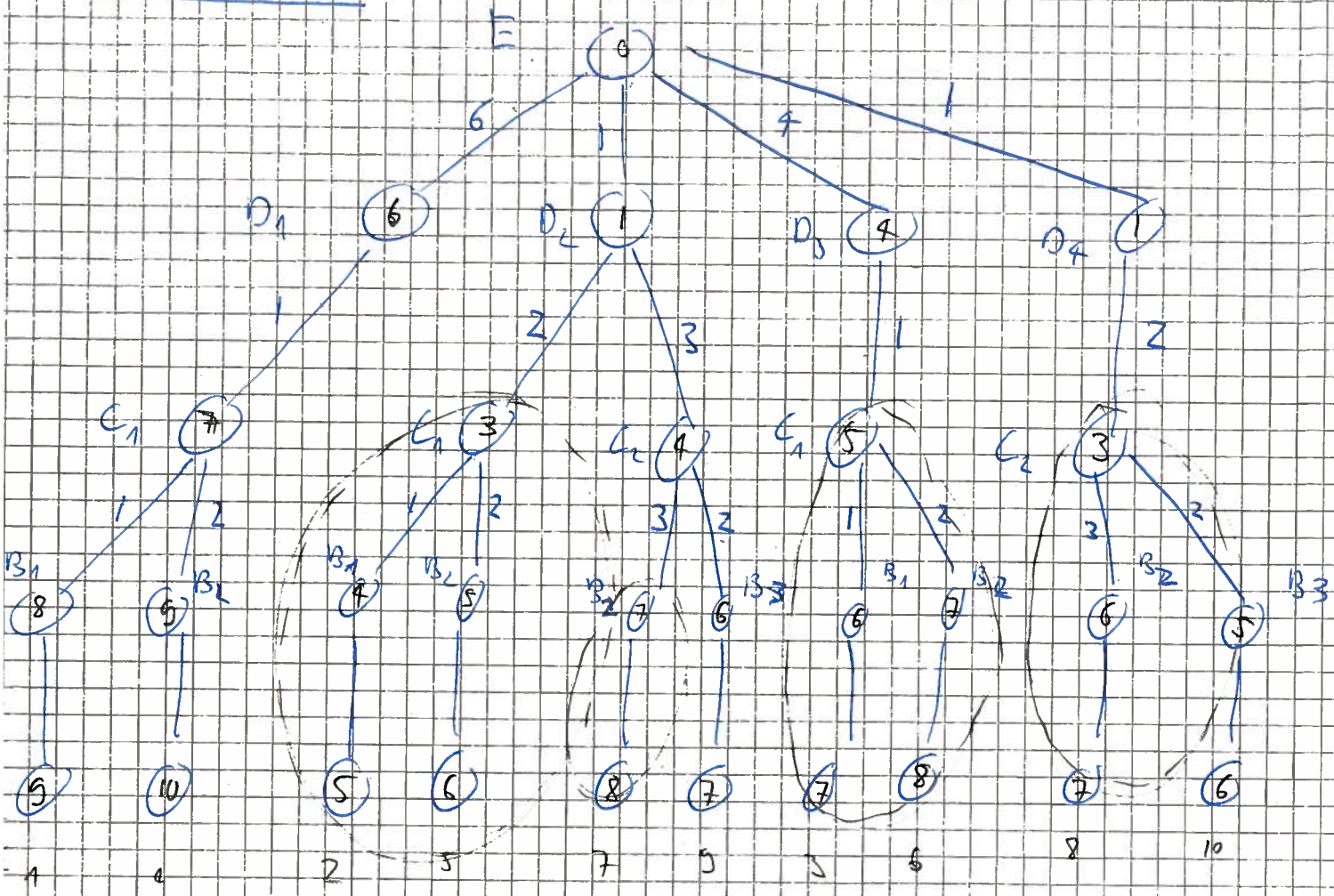
Backwards search ⟹ Decision tree starting at E

⇒ With pruning: same # computations as forward w. pruning

BUT Computed "cost" at nodes is now giving a "control" strategy !

⇐ Value function

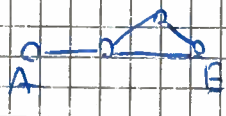
Backward Search

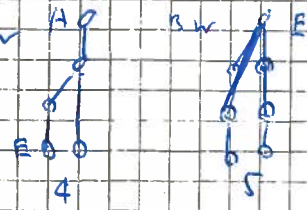


⇒ Now I have the info to make decisions
⇒ Just look for path with smallest number

⇒ This numbers are the value function!
What if I took a wrong turn? ⇒ I have info about path from all nodes (e.g. can fill in forward tree)

⇒ # Edges 29 ⇐ backward / fw decision tree are NOT identical





BUT w priming ⇒ 16 Evaluations

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

OLCAR Lecture #2 - 24.9.13

Principle of Optimality [Be] p18/19

From shortest path example we realize:

=> Can construct optimal path by starting at the „tail", always extending one step at the time

⇐ If path ABCD is optimal
⇔ BCD is also optimal
(or use controls; general: $u_0^* \dots u_N^*$ optimal
⇔ $u_1 \dots u_N$ optimal

=> Lets us formulate the task of finding an optimal path as a recursive problem!

=> See implications for more formal description → general
of the problem using definition of cost functions

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

OLCAR Lecture #2 24.9.13

Bellman Equation

Discrete time, Deterministic System

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

Cost function

$$V(t_n, x_n) = \underbrace{\phi(x_{t_f})}_{\text{Final cost, Terminal cost}} + \sum_{k=n}^{n_f-1} L(x_k, u_k)$$

recursion

Principle of optimality applied

$$\Rightarrow V^*(t_n, x_n) = \min_{u_n} \left(L(x_n, u_n) + V^*(u_{n+1}) \right)$$

Bellman Equation

optimal cost, value function (cf. backwards search example B)

$\Rightarrow$ Functional Equation (what's $V(t)$?)

Notation [Be]

$$J = V$$
$$J^* = V^*$$

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

OLCAR Lecture #2 24.9.13

Discrete time, stochastic

$$x_{n+1} = \underbrace{f(x_n, u_n)}_{\text{deterministic}} + \underbrace{w_k}_{\text{Random number}} \qquad w_k \sim P_w(\cdot \mid x_n, u_n)$$

or

$$x_{n+1} \sim P_f(\cdot \mid x_n, u_n)$$

$$V^*(t_0, x_0) = \min_{u_{0 \to t_f}} \left\{ \phi(x_{t_f}) + \sum_{k=t_0}^{t_f} L(x_k, u_k) \right\}$$

recursion!

$$V^*(n, x) = \min_u \left\{ L(n, u_n) + \mathop{E}_{x' \sim P(\cdot \mid x, u)} \left[V^*(n+1, u) \right] \right\}$$

Bellman Equation

Note!
- minimization on single u, not control sequence
- V function of time
- V is defined for complete state space

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

OLCAR Lecture #2 24.9.13

Note: In deterministic case $P(\cdot \mid x,u)$ become dirac delta

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

$$\Leftrightarrow V^*(n,u) = \min_{u_n} \left\{ L(n,u) + V^*(n+1, f(x_n,u)) \right\}$$

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

OLCAR Lecture #2 24.9.13

Infinite horizon, stochastic

$$V(x_0) = \min \left\{ E\left[\sum_{k=0}^{\infty} \alpha^k L(x_k, u_k) \right] \right\}$$

$\alpha \in [0, 1]$ $\Rightarrow$ see [Be] p 403

$$V^*(x_0) = \min_{u_n} \left\{ L(x, u) + \alpha E\left[V^*(x') \right] \right\}$$

Note: - V is <u>not</u> a function of time

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

OLCAR Lecture #2 24.9.13

Informal derivation of Bellman Equation

$$V(k, x) = \min_{u_k \dots u_{k_f}} E\left\{\phi(x_{k_f}) + \sum_{l=k}^{k_f-1} \alpha^{l-k} L(x_l, u_l)\right\}$$

$$= \min_{u_k \dots u_{k_f}} E\left\{\phi(x_{k_f}) + L(x_k, u_k) + \alpha \sum_{l=k+1} \alpha^{l-(k+1)} L(x_l, u_l)\right\}$$

$$= \min_{u_k} \left\{\phi(x_{k_f}) + L(x_k, u_k) + \min_{u_{k+1} \dots u_{k_f}} E\left\{\alpha \sum \alpha^{l-(k+1)} L(x_l, u_l)\right\}\right\}$$

$$= \min_{u_k} \left\{L(x_k, u_k) + \alpha E\left[V(k+1, x')\right]\right\}$$

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

OLCAR Lecture #2 24.9.13

Hamilton Jacobi Bellman Equation

[Be] p 109/110

Continuous time, deterministic

$$V(x_0) = \phi(t_f) + \int_0^{t_f} L(x_t, u_t)\, dt$$

$$\dot{x} = f(x, u) \qquad x, u : f(t) \;?$$

Use discretization

$$t_{n+1} = t_n + \delta \qquad \delta = \frac{T}{N}$$

$$V(x_0) = \phi(x_{t_f}) + \sum_{k=0}^{t_f/\delta - 1} L(x_{k\delta t}, u_{k\delta t})\, \delta t$$

{ Taylor series of V

$$V(\delta(n+1),\, x') = V(\delta n, x) + \frac{\partial V}{\partial t}\delta t + \left(\frac{\partial V}{\partial x}\right)^T f(x,u)\delta t$$

$$-\frac{\partial V}{\partial t}\delta t = \min_{u_{n\delta t}} \left\{ L(u_{\delta t_n}, u_{n\delta t})\,\delta t + \left(\frac{\partial V}{\partial x}\right)^T f(x,u)\delta t \right.$$

(use $\delta t \neq 0$, remove)

$$-\frac{\partial V}{\partial t} = \min_u \left\{ L(x,u) + \left(\frac{\partial V}{\partial x}\right)^T f(x,u) \right\}$$

Infinite time $\min_u \left\{ L(x,u) + \frac{\partial V}{\partial x})^T f(x,u) \right\} = 0$

$\Rightarrow V \neq f(t)$, but $V = f(x)$!

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

OLCAR Lecture #2 24.9.13

Stochastic HJB

$$-\frac{\partial V}{\partial t} = \min_u \left(L(x,u) + (\nabla_x V)^T f + \frac{1}{2} \text{trace}\, \nabla_{xx} V G \Sigma G^T \right)$$

$$dx = F(x,u)dt + G\,dw$$

Σ Noise covariance Matrix

infinite time $\partial_t V = 0$