Linear Complementarity for Regularized Policy Evaluation and Improvement

Jeff Johns *Christopher Painter-Wakefield* *Ronald Parr*

Duke University

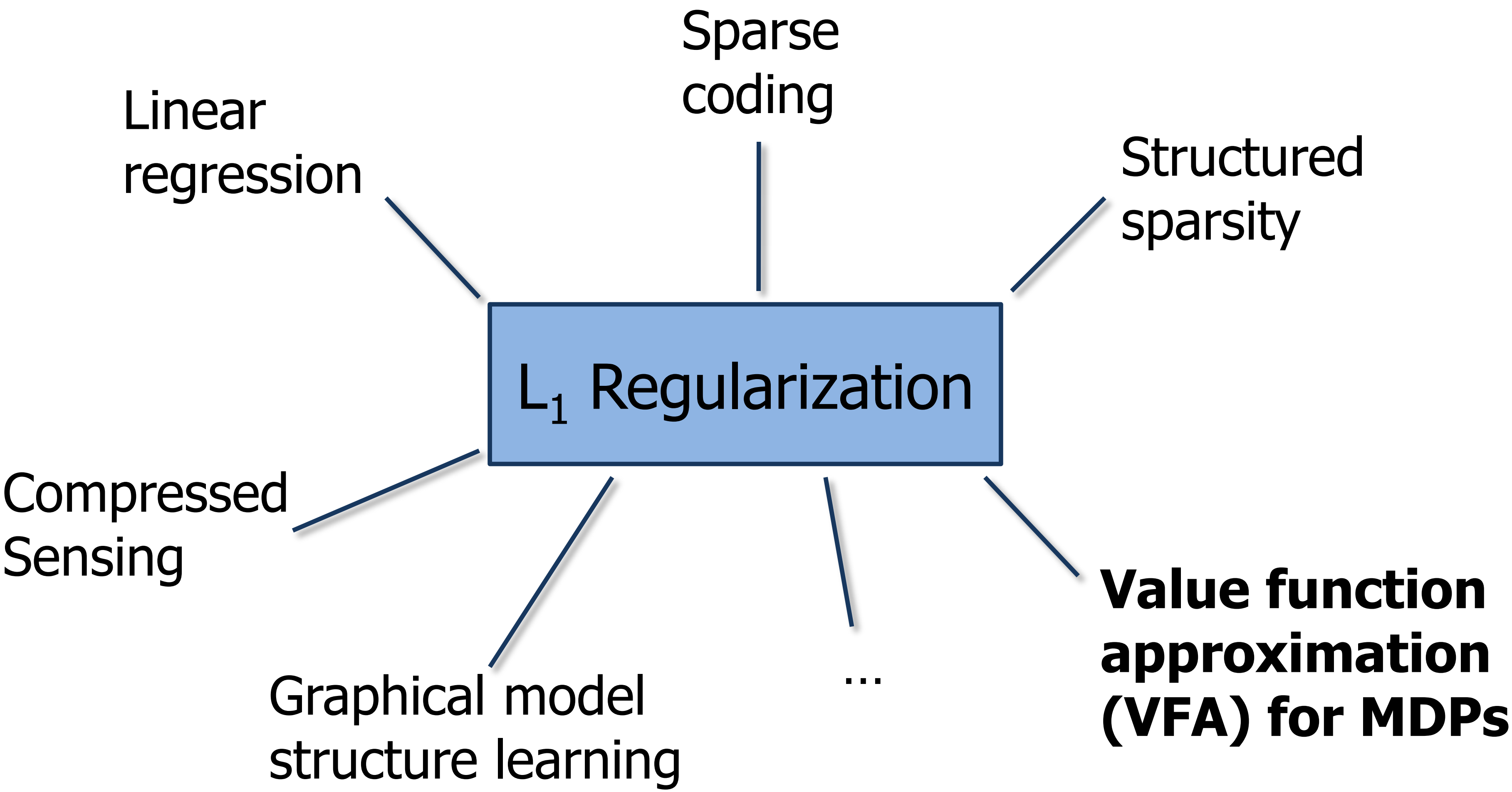
This Talk

What	Solving multiple, related L_1 regularization problems
Why	Efficient policy iteration algorithm for Markov decision processes (MDPs)
How	Formulate as a linear complementarity problem (LCP), use warm starts

L_1 Regularization

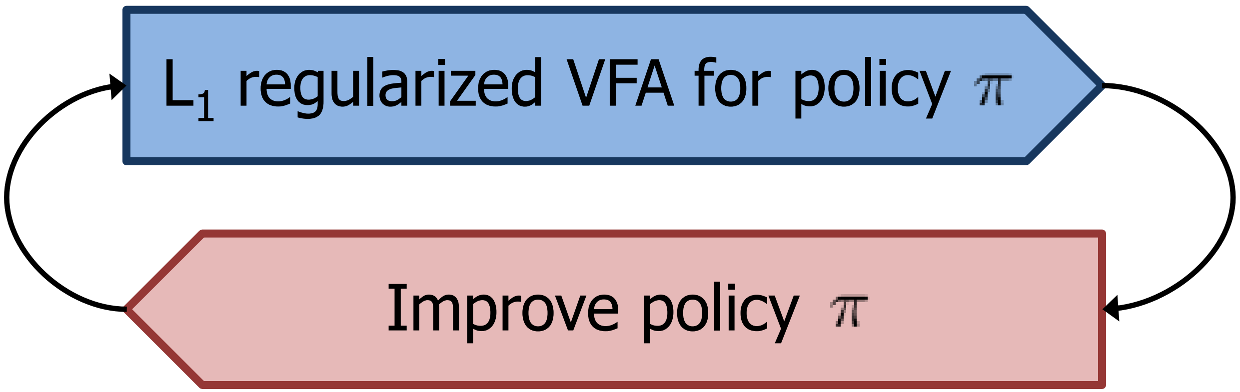
- Form
  - Optimization: $w = \operatorname*{argmin}_u \left(L(u) + \beta \|u\|_1\right)$
  - Fixed point: $w = \operatorname*{argmin}_u \left(L(u, w) + \beta \|u\|_1\right)$
- Benefits
  - *Sparse* solutions (model interpretability)
  - Learn with large number of irrelevant features
  - Prevents overfitting

Applications



VFA for MDPs

- Policy iteration



- Role of linear complementarity
  - New VFA method
  - Warm starts accelerate computation across iterations

Agenda

- ✔ Big Picture
- Background
- Connection to LCPs
- Two LCP-based Algorithms
- Summary

Markov Decision Processes

- Finite MDP: $\langle S, A, P, R \rangle$
- Goal: Learn a "good" policy $\pi : S \rightarrow A$
- Value function

$$V^\pi(s) = R(s) + \gamma \sum_{s' \in S} P_{ss'}^{\pi(s)} V^\pi(s')$$

$$V^\pi = R + \gamma P^\pi V^\pi \equiv T^\pi(V^\pi)$$

Linear VFA

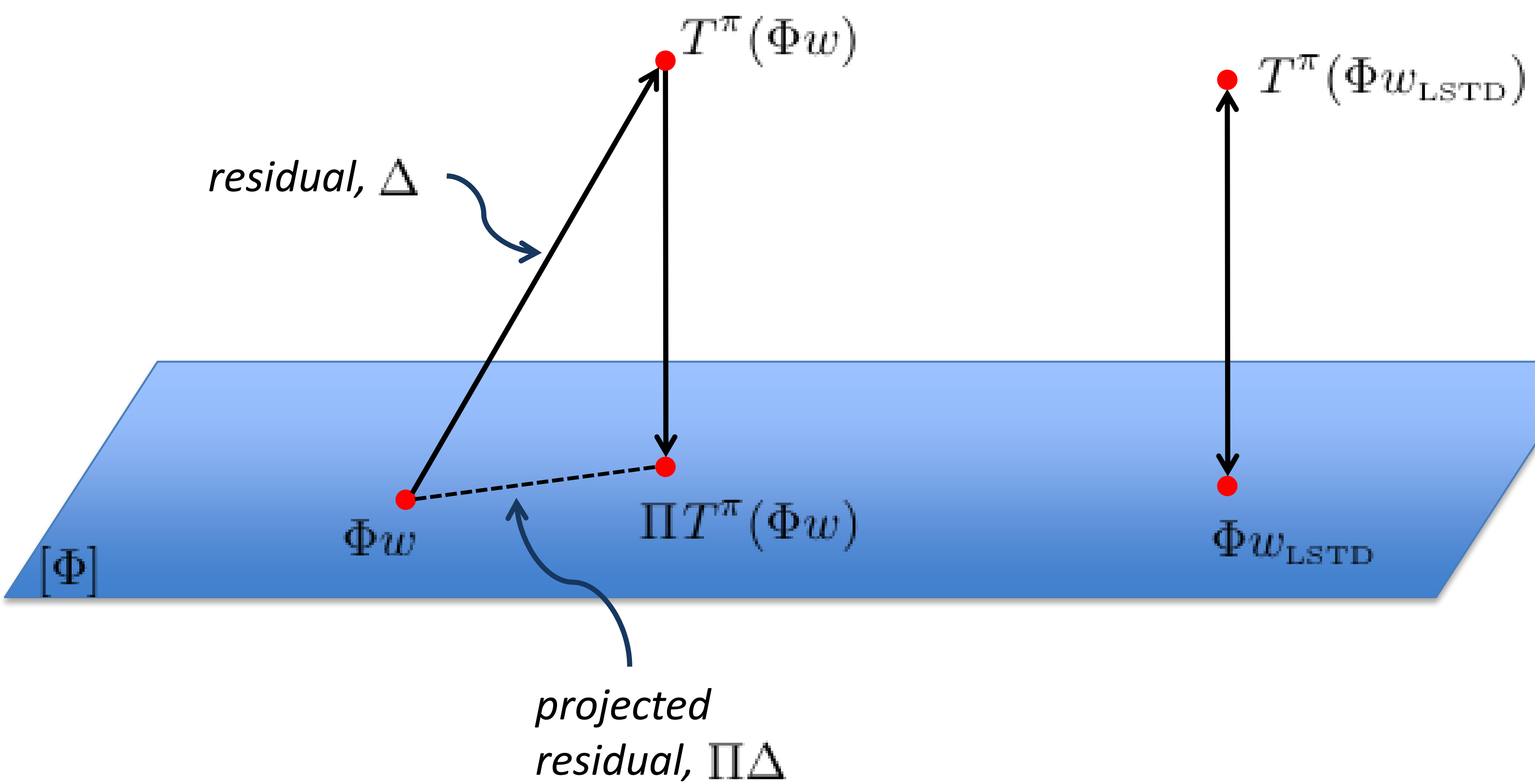
- Linear approximation

$$\hat{V}^\pi(s) = \sum_j \phi_j(s) w_j$$
$$\hat{V}^\pi = \Phi w$$

- Set w by considering residual $\Delta = T^\pi(\Phi w) - \Phi w$

LSTD [Bradtke & Barto, '96]

- Set w to minimize projection of the residual



L_1TD [Kolter & Ng, '09]

- L_1TD = LSTD + L_1 regularization

- L_1 fixed point

$$w = \operatorname*{argmin}_u \left(L(u,w) + \beta \|u\|_1 \right)$$

$$L(u,w) = \frac{1}{2}\|\Phi u - T^\pi(\Phi w)\|_2^2$$

L_1TD Solution

- No analytical solution
- Coefficients must meet first order conditions

Derivation parallels LASSO conditions

$$c = \Phi^T \Delta$$

$$
\begin{aligned}
-\beta \leq c_i \leq \beta & \qquad \forall i \\
c_i = \beta & \;\Rightarrow\; w_i \geq 0 \\
c_i = -\beta & \;\Rightarrow\; w_i \leq 0 \\
-\beta < c_i < \beta & \;\Rightarrow\; w_i = 0
\end{aligned}
$$

$\beta \geq 0$, *L_1 reg. parameter*

LARS-TD [Kolter & Ng, '09]

- Solves L_1TD problem
- Homotopy method inspired by LARS [Efron et al., '03]
  - Gives solution for *all* values of β
- Disadvantages w.r.t. policy iteration
  - Each policy starts from scratch
  - Must commit to one β to perform policy improvement

Agenda

✔ • Big Picture

✔ • Background

• Connection to LCPs

• Two LCP-based Algorithms

• Summary

Linear Complementarity Problems (LCPs)

- LCPs are mathematical programs
  - LP ⊂ LCP ⊂ QP

- Given: q, M (square)

  Compute: d, x

$$d = Mx + q$$

$$d \geq \mathbf{0}, \;\; x \geq \mathbf{0}$$

$$d_i x_i = 0 \qquad \forall i$$

L_1TD as an LCP

- Recall $c = \Phi^T\Delta = \underbrace{\Phi^T R}_{b} - \underbrace{\Phi^T(\Phi - \gamma P^\pi \Phi)}_{A} w$

- LCP inputs:

$$q = \left(\beta + \begin{bmatrix} b \\ -b \end{bmatrix}\right)$$

$$M = \begin{bmatrix} A & -A \\ -A & A \end{bmatrix}$$

- LCP solution achieves L_1TD first order conditions
  - Denote this as $w \leftarrow \text{LC-TD}(A, b, \beta)$

LCP Solvers

- Similarities to LP solvers
  - Worst case exponential complexity

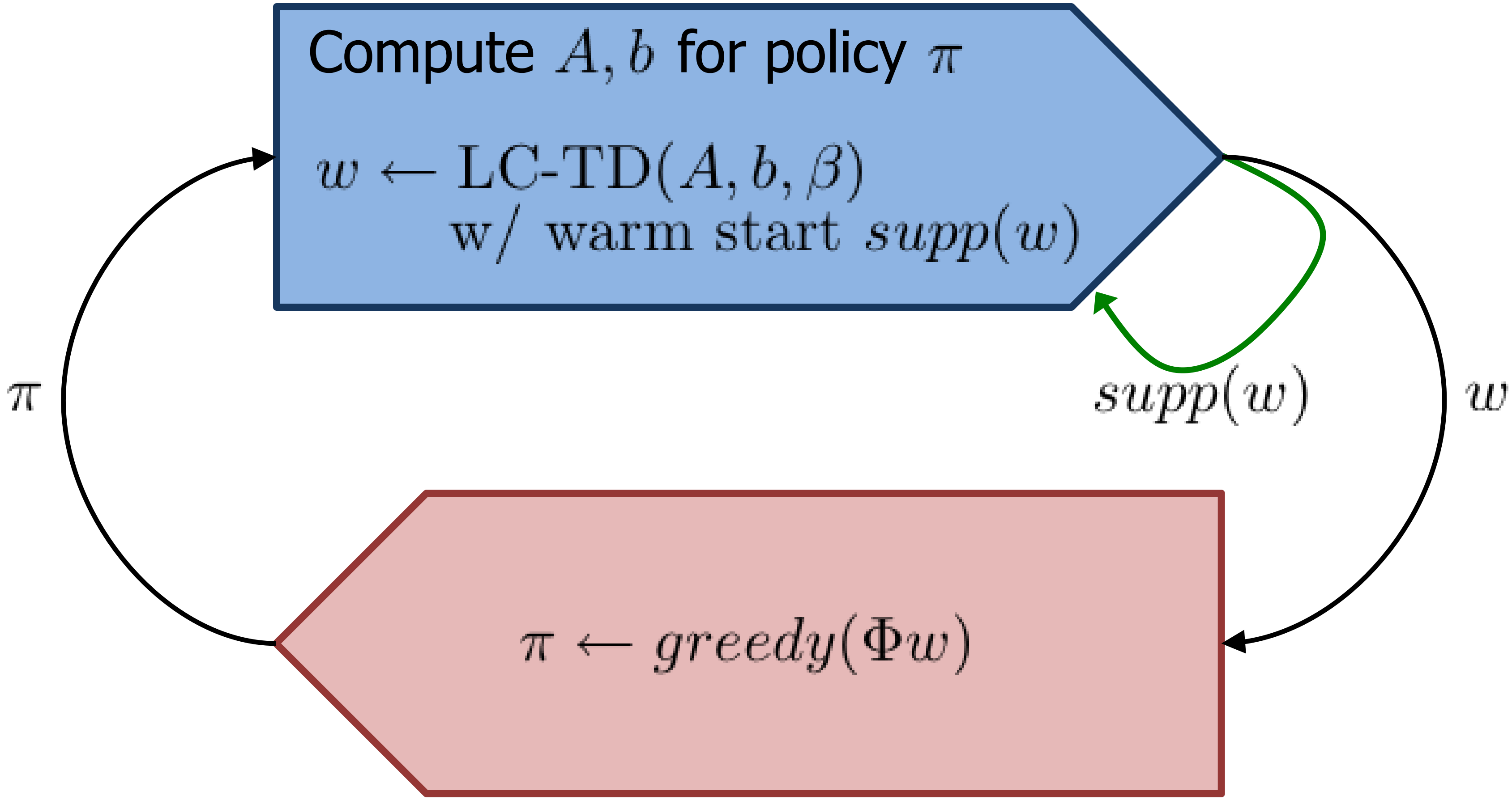
- Some solvers can be initialized with a *warm start*

- Properties of q, M dictate solution existence, uniqueness, and computability

Agenda

- ✔ Big Picture
- ✔ Background
- ✔ Connection to LCPs
- Two LCP-based Algorithms
- Summary

Algorithm 1: LC-TD w/ PI

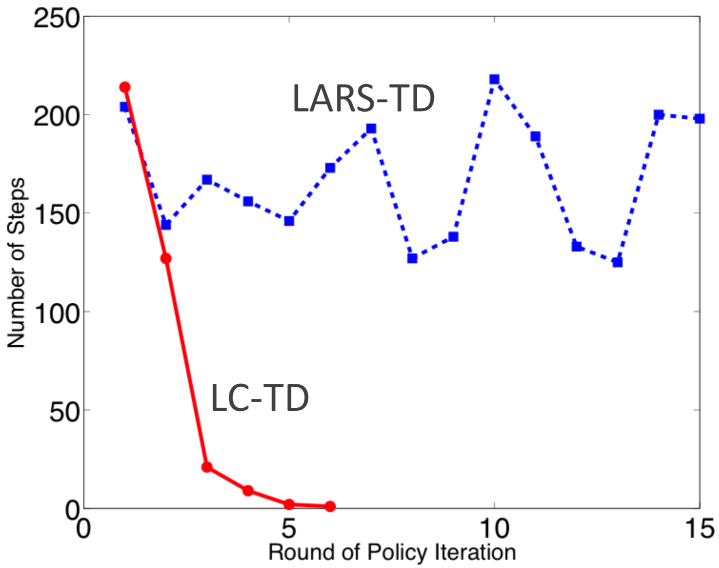


Experiments

- Compare average number of algorithm steps for LC-TD and LARS-TD

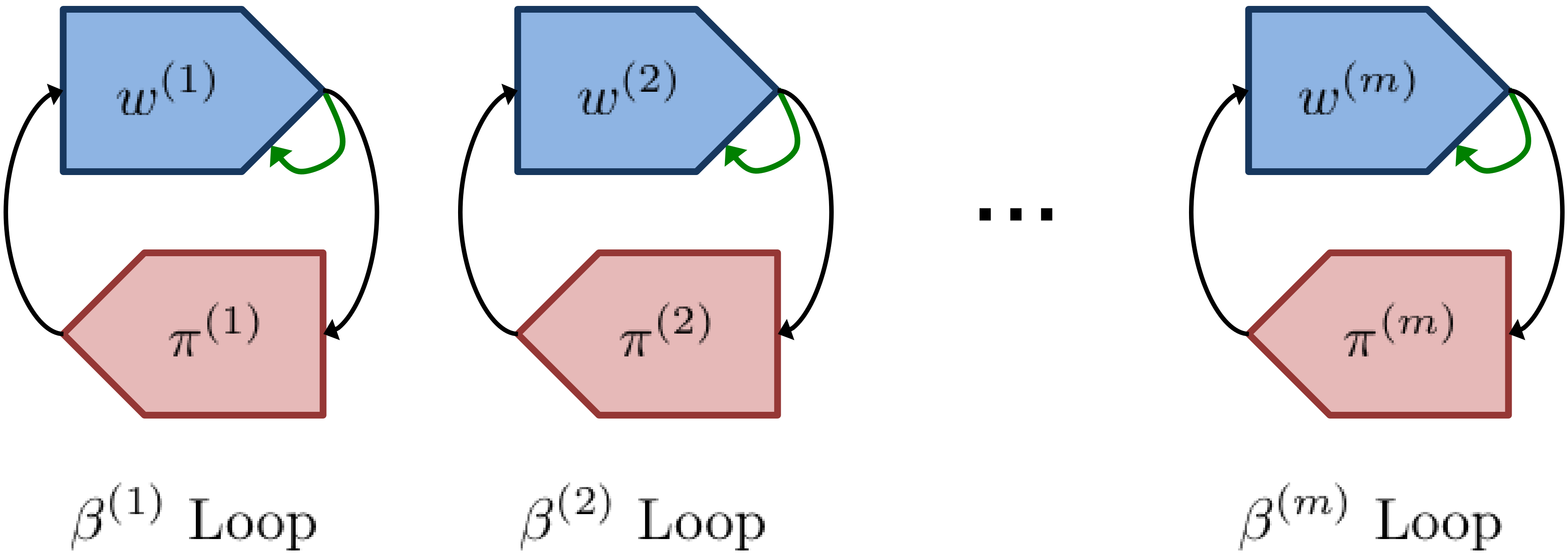
Domain	LARS-TD, PI	LC-TD, PI
Mountain Car	214 ± 23	**116 ± 22**
Chain	73 ± 13	77 ± 11
Pendulum	153 ± 25	**48 ± 23**

Mountain Car MDP



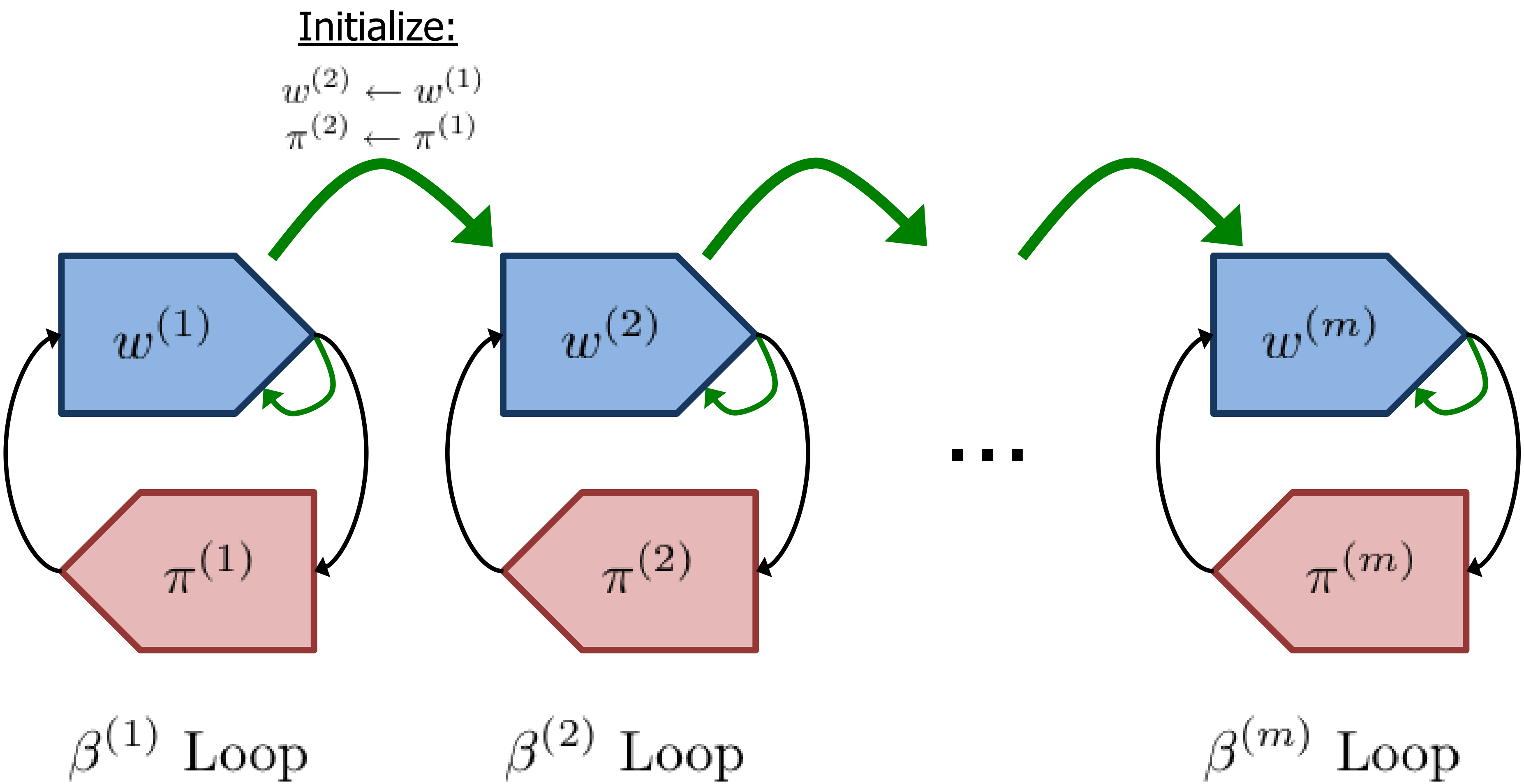
Choosing β

- Option 1: Run *m* separate trials



Algorithm 2: LC-MPI

Let $\beta^{(1)} > \beta^{(2)} > \ldots > \beta^{(m)}$



Experiments

- Compare avg. number of algorithm steps
  - LARS-TD and LC-TD 11 separate times for different β's
  - LC-MPI once for same 11 β's

Domain	LARS-TD, PI	LC-TD, PI	LC-MPI
Mountain Car	214 ± 23	116 ± 22	**21 ± 5**
Chain	73 ± 13	77 ± 11	**24 ± 11**
Pendulum	153 ± 25	48 ± 23	**33 ± 18**

Agenda

- ✔ Big Picture
- ✔ Background
- ✔ Connection to LCPs
- ✔ Two LCP-based Algorithms
- Summary

Summary

- L_1 fixed point is equivalent to an LCP
- LCP formulation useful when solving multiple, related problems
  - Exploit warm starts
  - Reduce computation during policy iteration
  - LC-MPI lends itself to cross-validation

Thanks