Partitioned Linear Programming Approximations for MDPs

Branislav Kveton
Intel Research
Santa Clara, CA

Milos Hauskrecht
Department of Computer Science
University of Pittsburgh

Overview

- Introduction
  - Factored Markov decision processes
  - Approximate linear programming
  - Solving ALP formulations
- Partitioned linear programming approximations
  - Formulation, theory, and insights
- Experiments
- Conclusions and future work

Overview

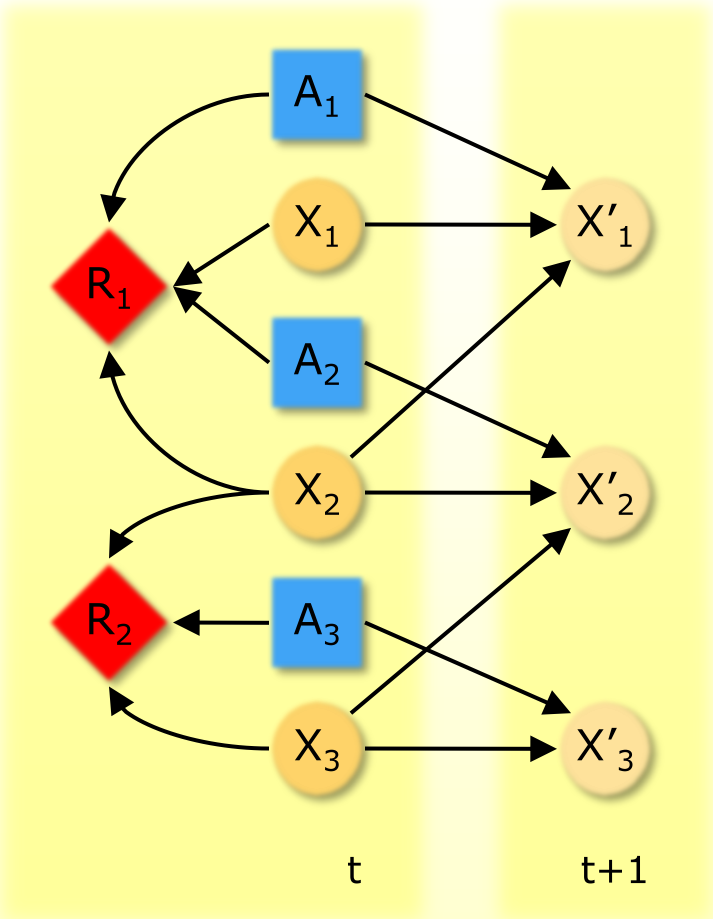
- Introduction
  - Factored Markov decision processes
  - Approximate linear programming
  - Solving ALP formulations
- Partitioned linear programming approximations
  - Formulation, theory, and insights
- Experiments
- Conclusions and future work

Factored Markov Decision Processes

- A factored Markov decision process (MDP) is a 4-tuple M = (**X**, A, P, R):
  - **X** is a set of state variables
  - A is a set of actions
  - P is a transition function represented by a dynamic Bayesian network (DBN)
  - R is a reward model:

$$R(\mathbf{x}, a) = \sum_j R_j(\mathbf{x}, a)$$

Local reward functions



Linear Value Function Approximations

- The quality of a policy is measured by the infinite horizon discounted reward:

$$E_\pi\left[\sum_{t=0}^{\infty}\gamma^t R(\mathbf{x}_t, \pi(\mathbf{x}_t))\right]$$

- The optimal value function V^* is a fixed point of the Bellman equation:

$$V^*(\mathbf{x}) = \max_a\left[R(\mathbf{x}, a) + \gamma E_{P(\mathbf{x}'|\mathbf{x},a)}\left[V^*(\mathbf{x}')\right]\right]$$

- A compact representation of an MDP may not guarantee a compact form of the optimal value function V^*
- Approximation of V^* by a linear combination of basis functions [Bellman *et al.* 1963, Van Roy 1998]:

$$V^{\mathbf{w}}(\mathbf{x}) = \sum_i w_i f_i(\mathbf{x})$$

Local feature functions

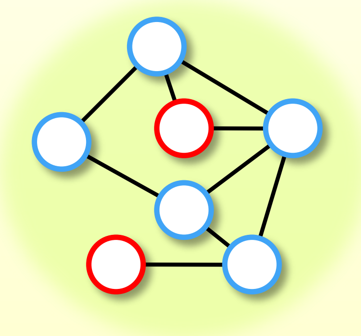
Approximate Linear Programming

- Optimization of the linear value function approximation can restated as an approximate linear program (ALP):

$$\begin{aligned} &\text{minimize}_{\mathbf{w}} \quad \mathrm{E}_{\psi}\left[\mathrm{V}^{\mathbf{w}}\right] \\ &\text{subject to:} \quad \mathrm{V}^{\mathbf{w}}(\mathbf{x}) \geq \mathrm{T}^{*}\mathrm{V}^{\mathbf{w}}(\mathbf{x}) \quad \forall \mathbf{x} \in \mathbf{X} \end{aligned}$$

- The linear value function approximation combined with the structure of factored MDPs induces a structure in ALP:

$$\begin{aligned} &\text{minimize}_{\mathbf{w}} \quad \sum_{i} w_i \alpha_i \\ &\text{subject to:} \quad \sum_{i} w_i \mathrm{F}_i(\mathbf{x}, a) - \sum_{j} \mathrm{R}_j(\mathbf{x}, a) \geq 0 \\ &\qquad \forall \mathbf{x} \in \mathbf{X}, a \in \mathrm{A} \end{aligned}$$



Constraint space of an ALP represented by a cost network

State-of-the-Art Methods for ALP

- Exact methods
  - Rewrite constraint space compactly (Guestrin *et al.* 2001)
  - Cutting plane method (Schuurmans & Patrascu 2002):

$$\arg\max_{\mathbf{x},a}\left\{\sum_i w_i^{(t)}\,\mathrm{F}_i(\mathbf{x},a)-\sum_j \mathrm{R}_j(\mathbf{x},a)\right\}$$

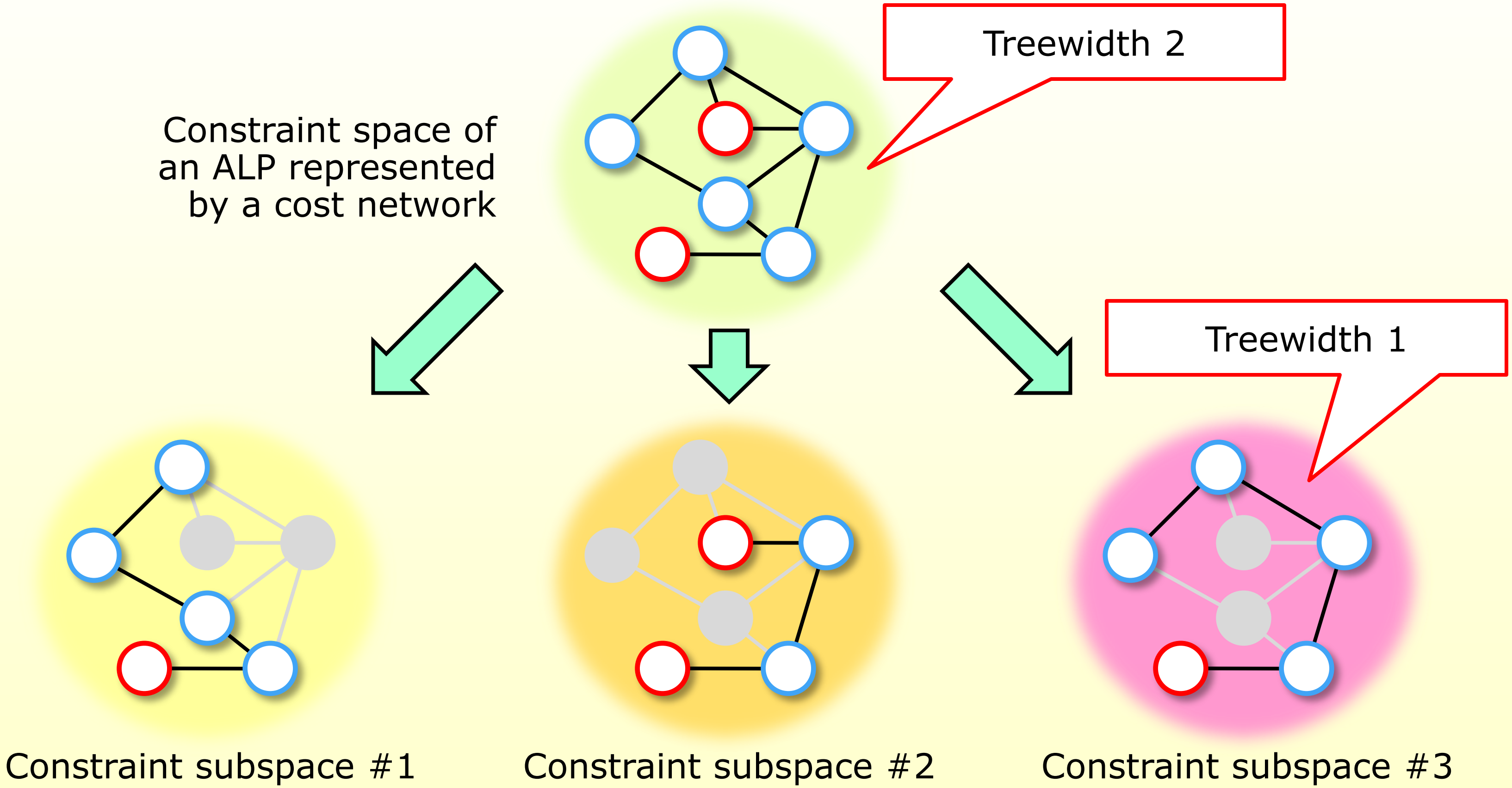
  - Problem: Exponential in the treewidth of the dependency graph that represents the constraint space in ALP
- Approximate methods
  - Monte Carlo constraint sampling (de Farias & Van Roy 2004)
  - Markov Chain Monte Carlo (MCMC) constraint sampling (Kveton & Hauskrecht 2005)
  - Problem: Stochastic nature and slow convergence in practice

Overview

- Introduction
  - Factored Markov decision processes
  - Approximate linear programming
  - Solving ALP formulations
- Partitioned linear programming approximations
  - Formulation, theory, and insights
- Experiments
- Conclusions and future work

Partitioned ALP Approximations

- Decompose the ALP constraint space (with a large treewidth) into a set of constraint subspaces (with small treewidths)



Partitioned ALP Approximations

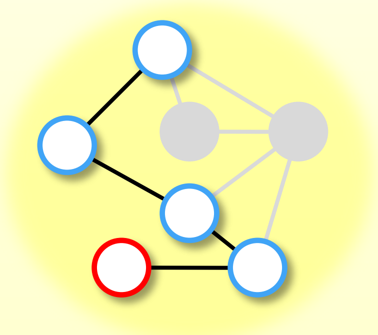
- Partitioned ALP (PALP) formulation with K constraint spaces is given by a linear program:

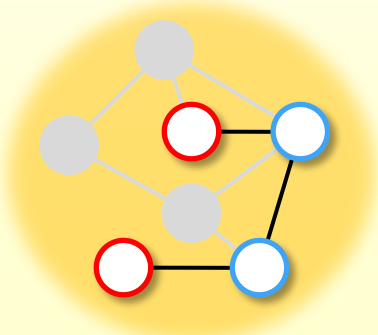
Partitioning matrix **D**

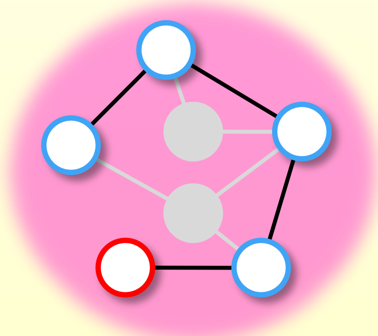
Column vector $\mathbf{M_w}(\mathbf{x}, a)^T$ of instantiated cost network terms

$$\text{minimize}_{\mathbf{w}} \quad \sum_i w_i \alpha_i$$

$$\text{subject to:} \quad \begin{pmatrix} d_{1,1} & d_{1,2} & d_{1,3} & \cdots \\ d_{2,1} & d_{2,2} & d_{2,3} & \cdots \\ d_{3,1} & d_{3,2} & d_{3,3} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} \mathrm{F}_1(\mathbf{x}, a) \\ \vdots \\ \mathrm{R}_1(\mathbf{x}, a) \\ \vdots \end{pmatrix} \geq \mathbf{0} \quad \forall \mathbf{x} \in \mathbf{X}, a \in \mathrm{A}$$


Constraint subspace #1


Constraint subspace #2


Constraint subspace #3

Partitioned ALP Approximations

- Partitioned ALP (PALP) formulation with K constraint spaces is given by a linear program:

Partitioning matrix **D**

Column vector $\mathbf{M_w}(\mathbf{x}, a)^T$ of instantiated cost network terms

$$\text{minimize}_{\mathbf{w}} \quad \sum_i w_i \alpha_i$$

$$\text{subject to:} \quad \begin{pmatrix} d_{1,1} & d_{1,2} & d_{1,3} & \cdots \\ d_{2,1} & d_{2,2} & d_{2,3} & \cdots \\ d_{3,1} & d_{3,2} & d_{3,3} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} \mathrm{F}_1(\mathbf{x}, a) \\ \vdots \\ \mathrm{R}_1(\mathbf{x}, a) \\ \vdots \end{pmatrix} \geq \mathbf{0} \quad \forall \mathbf{x} \in \mathbf{X}, a \in \mathrm{A}$$

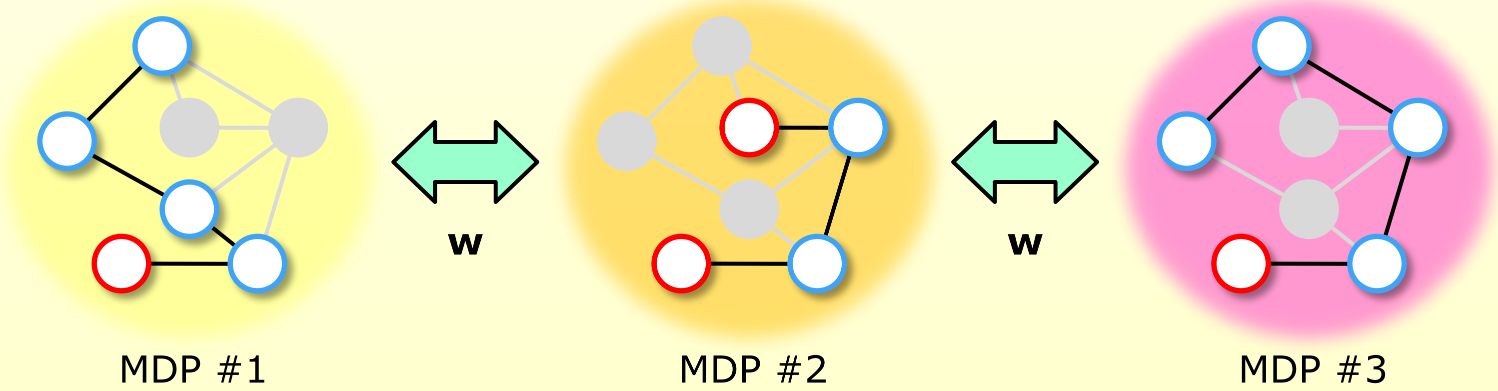
- When the decomposition **D** is convex, a solution to the PALP formulation is feasible in the corresponding ALP formulation
- The PALP formulation is feasible if the set of basis functions includes a constant basis function $f_0(\mathbf{x}) \equiv 1$

Interpreting PALP Approximations

- PALP can be viewed as solving K MDPs with overlapping state and action spaces, and shared value functions:

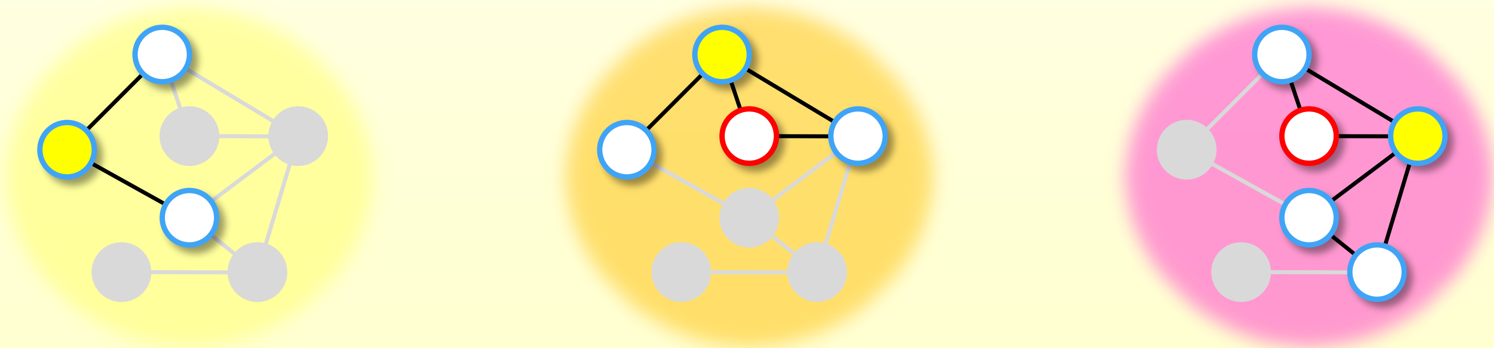
$$\text{minimize}_{\mathbf{w}} \quad \sum_i d_{1,i} w_i \alpha_i + \sum_i d_{2,i} w_i \alpha_i + \sum_i d_{3,i} w_i \alpha_i + \dots$$

$$\text{subject to:} \quad \begin{pmatrix} d_{1,1} & d_{1,2} & d_{1,3} & \cdots \\ d_{2,1} & d_{2,2} & d_{2,3} & \cdots \\ d_{3,1} & d_{3,2} & d_{3,3} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} \mathrm{F}_1(\mathbf{x}, a) \\ \vdots \\ \mathrm{R}_1(\mathbf{x}, a) \\ \vdots \end{pmatrix} \geq \mathbf{0} \qquad \forall \mathbf{x} \in \mathbf{X}, a \in \mathrm{A}$$



Partitioning Matrix D

- To achieve high quality and tractable approximations, the K constraint spaces should preserve critical dependencies in the MDP and have a small treewidth
- How to generate the best PALP approximation within a given complexity limit is an open question
- In the experimental section, we build a constraint space for every node in the ALP cost network and its neighbors



Constraint subspace #1 Constraint subspace #2 Constraint subspace #3

Solving PALP Approximations

- PALP formulations can be solved by exact methods for solving ALP formulations
- In the experimental section, we use the cutting plane method for solving linear programs

Theoretical Analysis

- PALP value functions are upper bounds on the optimal value function V^*
- PALP minimizes the L_1-norm error between the optimal value function V^* and our value function approximation
- The quality of PALP solutions can be bounded as follows:

$$\left\| V^* - V^{\tilde{\mathbf{w}}} \right\|_{1,\psi} \leq \frac{2}{1-\gamma} \min_{\mathbf{w}} \left\| V^* - V^{\mathbf{w}} \right\|_\infty + \frac{K\delta}{1-\gamma}$$

The L_1-norm error of the PALP value function

The minimum max-norm error of the linear value function approximation

The hardness of making an ALP solution feasible in the PALP formulation

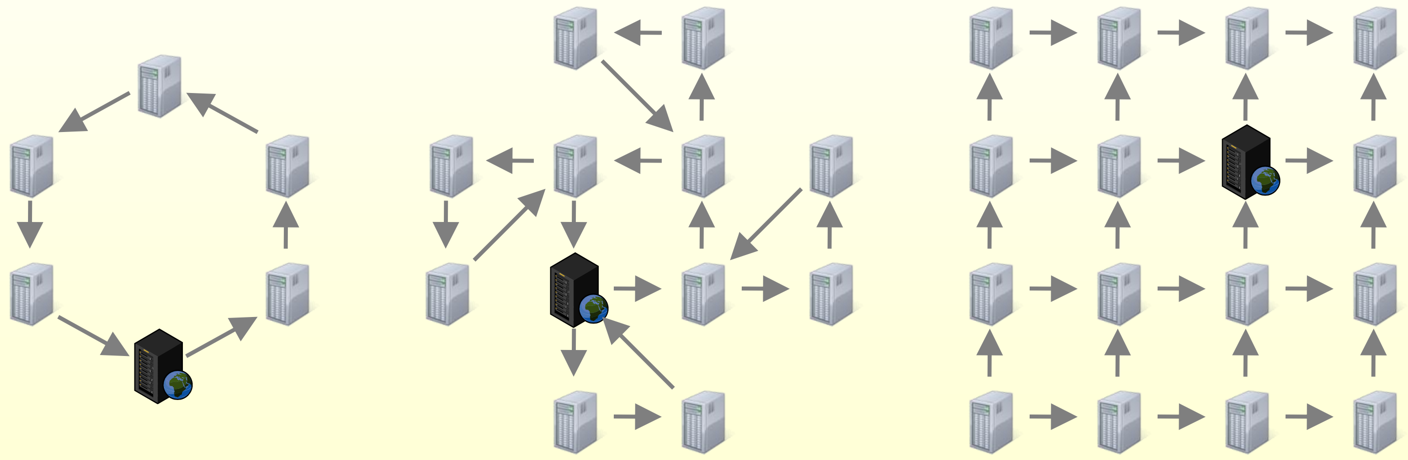
- PALP generates a close approximation to the optimal value function V^* if V^* lies in the span of basis functions and the penalty δ for partitioning the ALP constraint space is small

Overview

- Introduction
  - Factored Markov decision processes
  - Approximate linear programming
  - Solving ALP formulations
- Partitioned linear programming approximations
  - Formulation, theory, and insights
- Experiments
- Conclusions and future work

Experiments

- Demonstrate the quality and scale-up potential of partitioned ALP approximations
- Comparison to exact and Monte Carlo ALP approximations on three topologies of the network administration problem

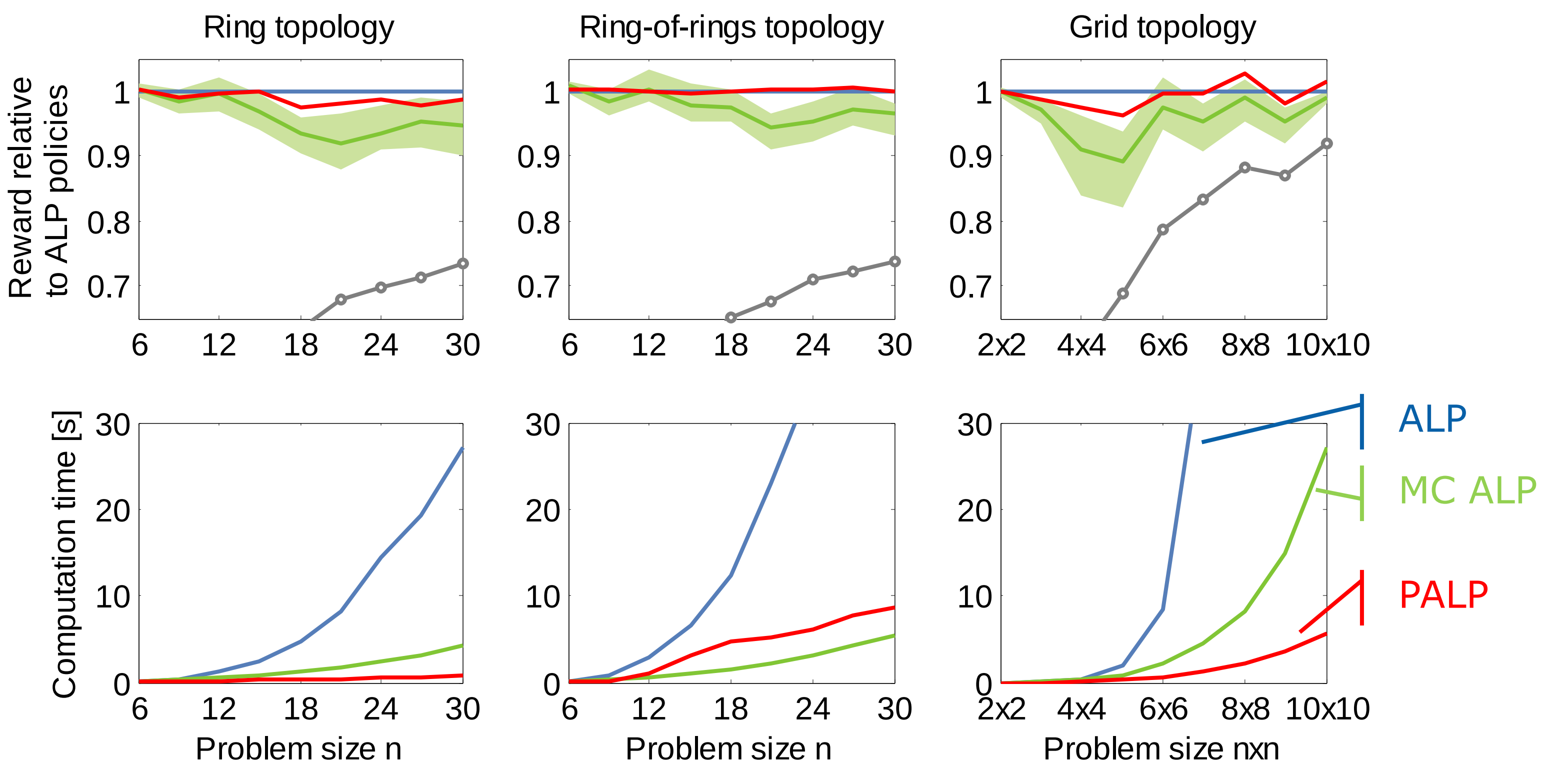


Ring topology Ring-of-rings topology Grid topology

Treewidth of the grid topology grows with the number of computers

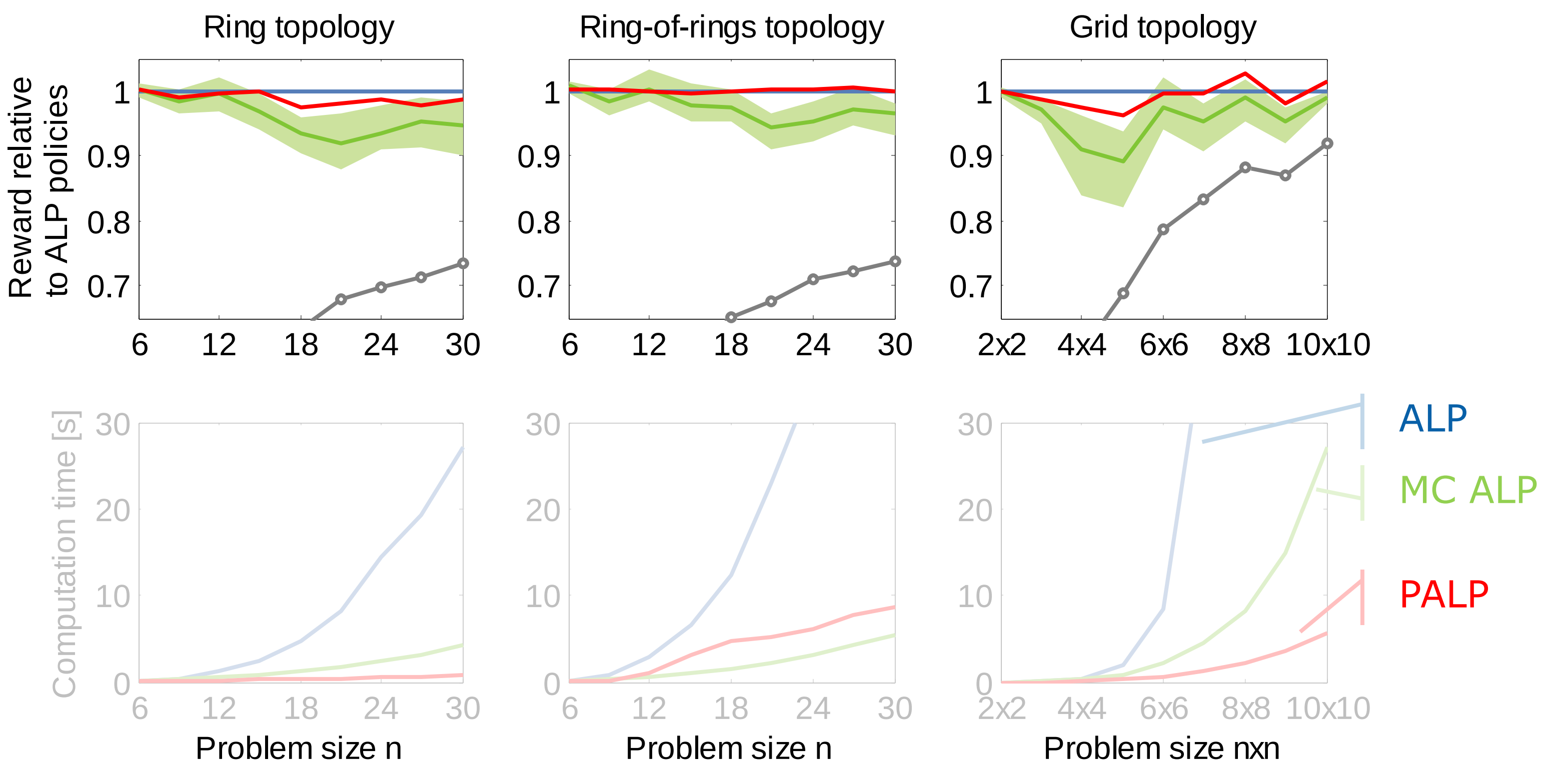
Experimental Results

- Evaluation by the quality of policies (relatively to the reward of ALP policies) and computation time



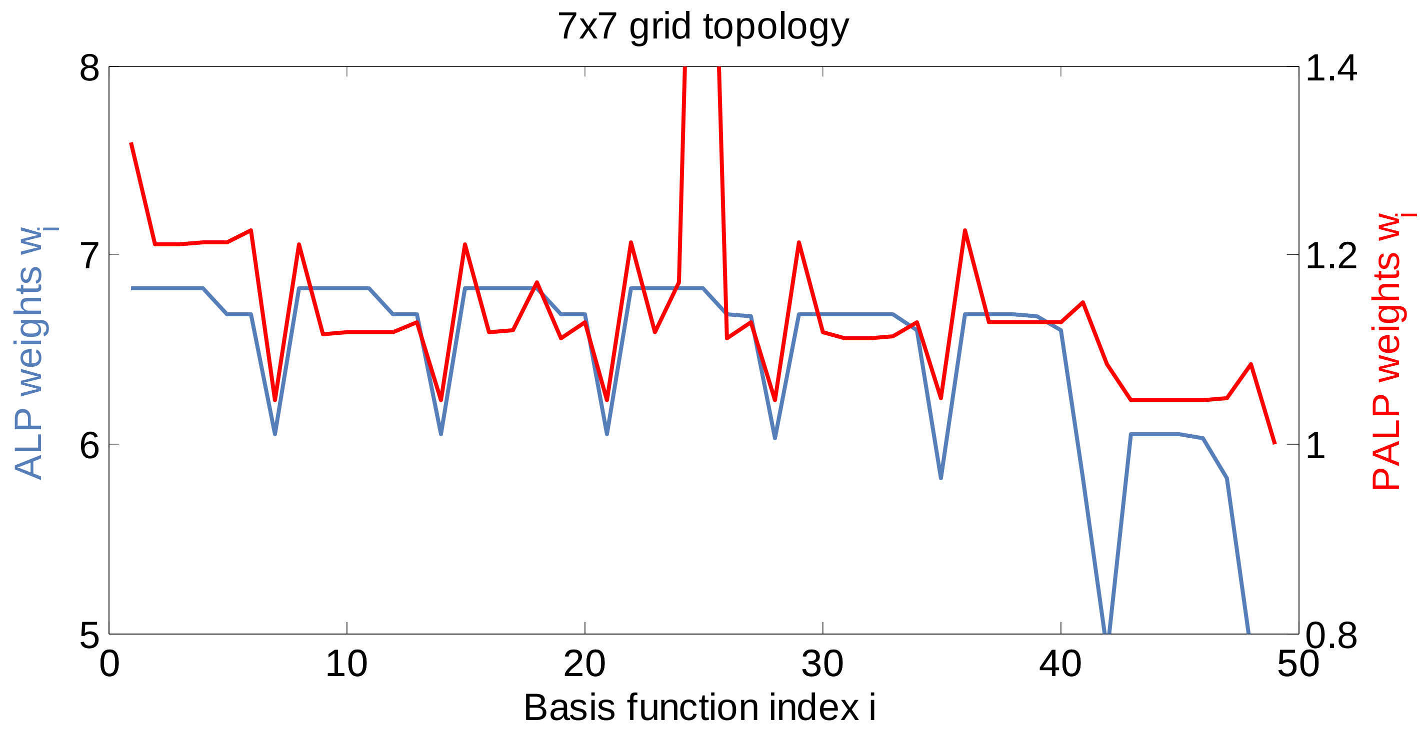
Experimental Results

- The quality of PALP policies is almost as high as the quality of ALP policies



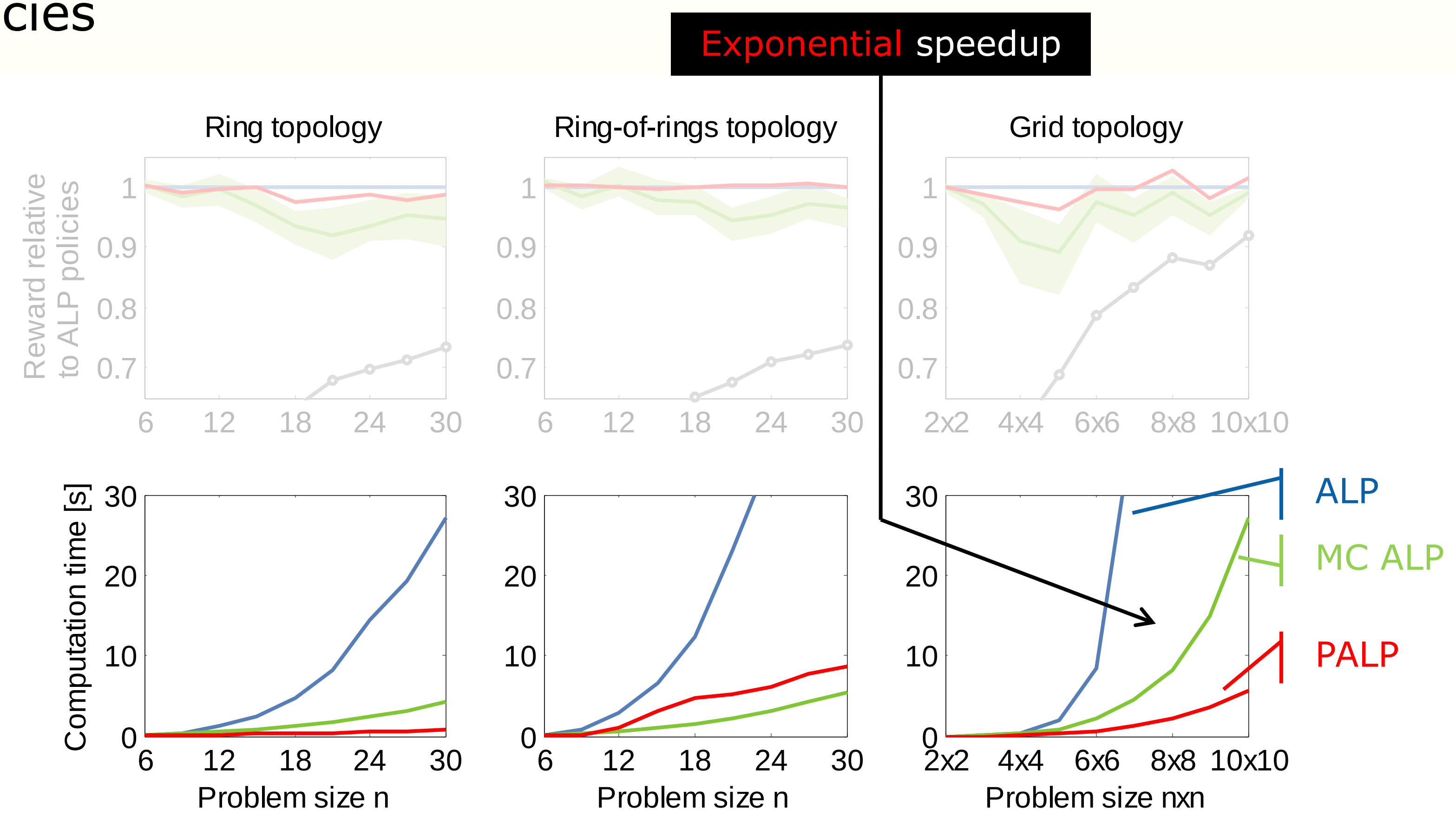
Experimental Results

- Magnitudes of ALP and PALP weights are different but the weights exhibit similar trends



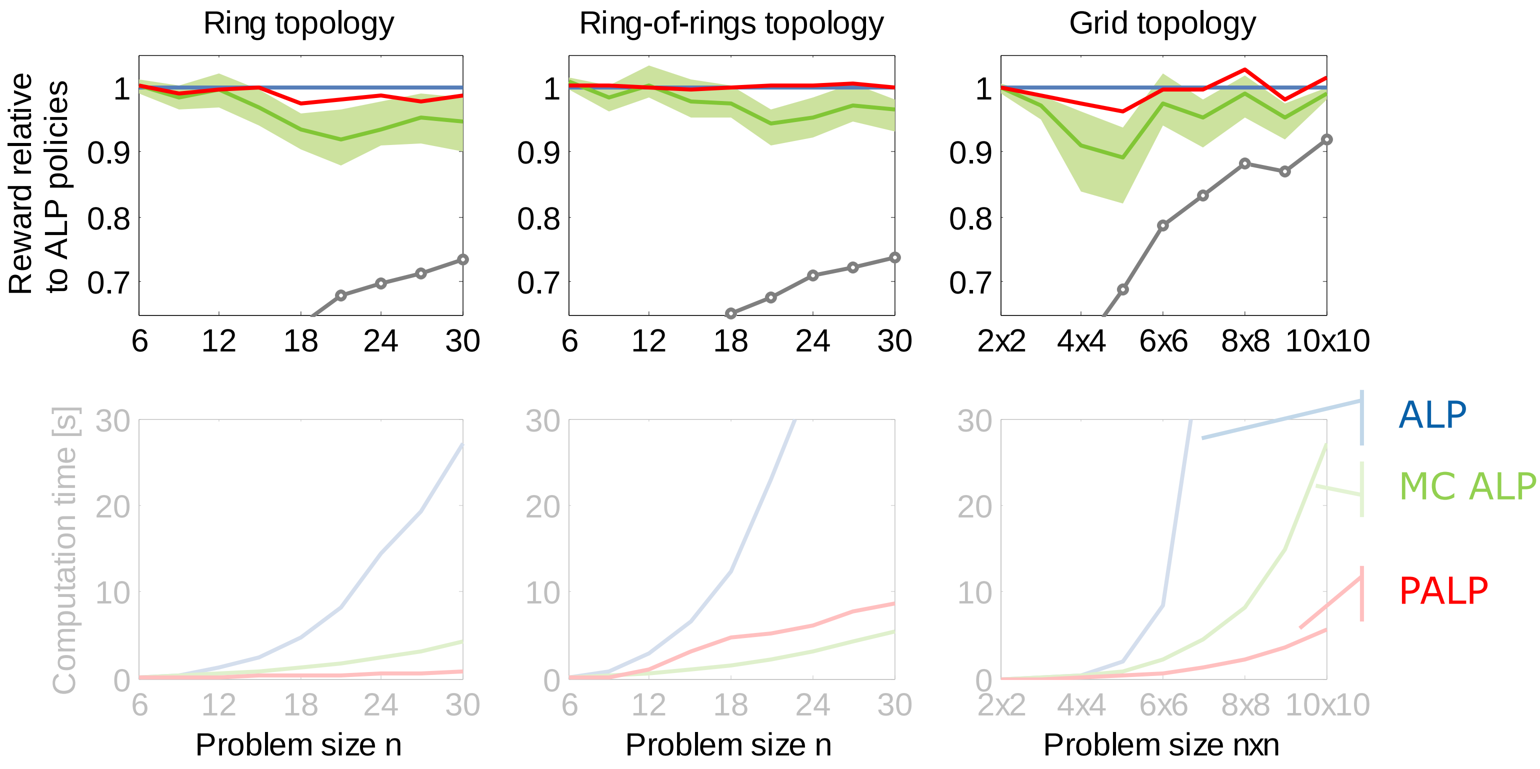
Experimental Results

- PALP policies can be computed significantly faster than ALP policies



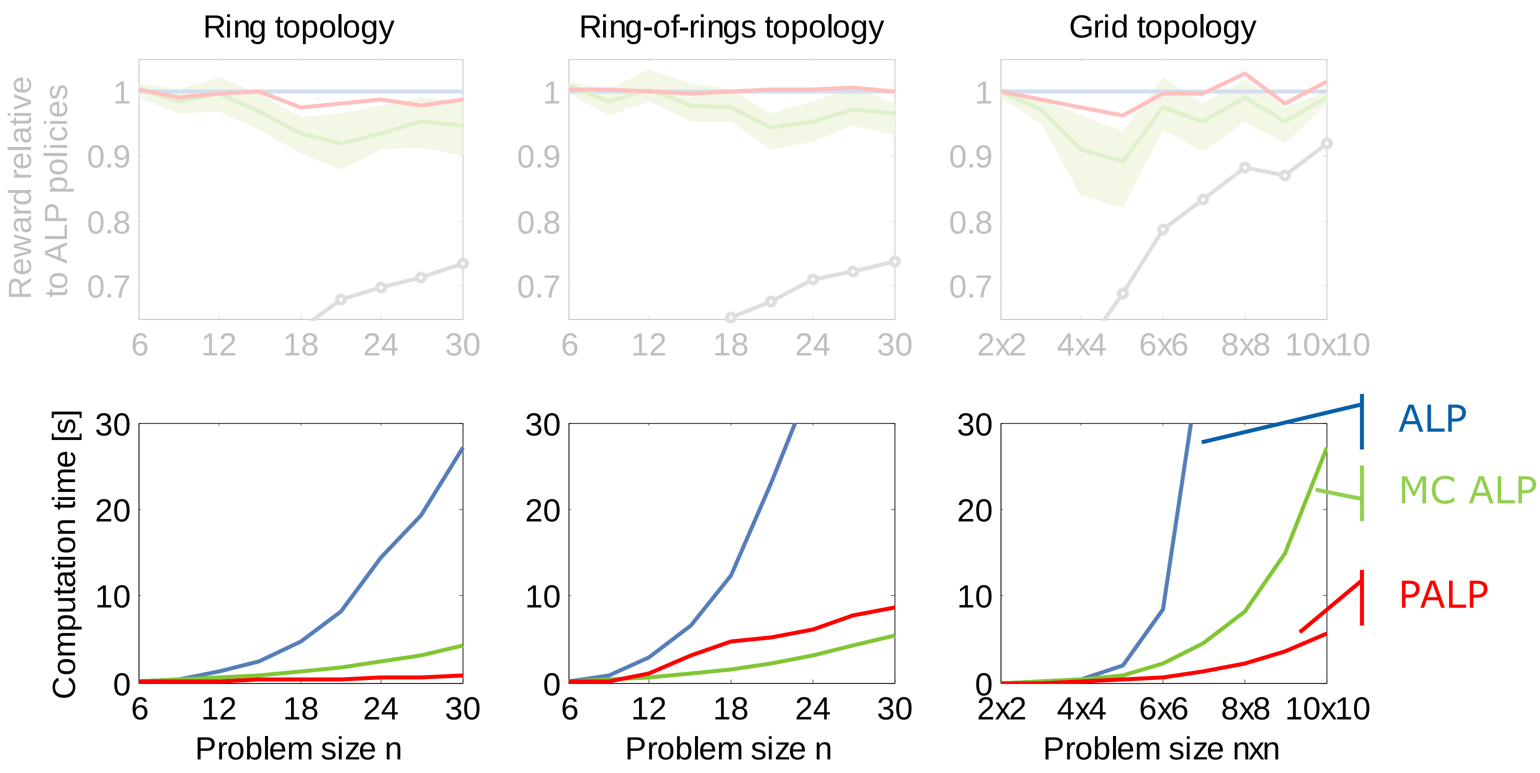
Experimental Results

- PALP policies are superior to ALP policies, which are obtained by Monte Carlo constraint sampling



Experimental Results

- PALP policies are superior to ALP policies, which are obtained by Monte Carlo constraint sampling



Overview

- Introduction
  - Factored Markov decision processes
  - Approximate linear programming
  - Solving ALP formulations
- Partitioned linear programming approximations
  - Formulation, theory, and insights
- Experiments
- Conclusions and future work

Conclusions and Future Work

- Conclusions
  - A novel approach to ALP that allows for satisfying ALP constraints without an exponential dependence on their treewidth
  - Natural tradeoff between the quality and computation time of ALP solutions
  - Bounds on the quality of learned policies
  - Evaluation on a challenging synthetic problem
- Future work
  - Learning of a good partitioning matrix **D** and the problem of exact inference in Bayesian networks with a large treewidth
  - Evaluate PALP on a large-scale real-world planning problem