Complex Backup Strategies in Monte Carlo Tree Search

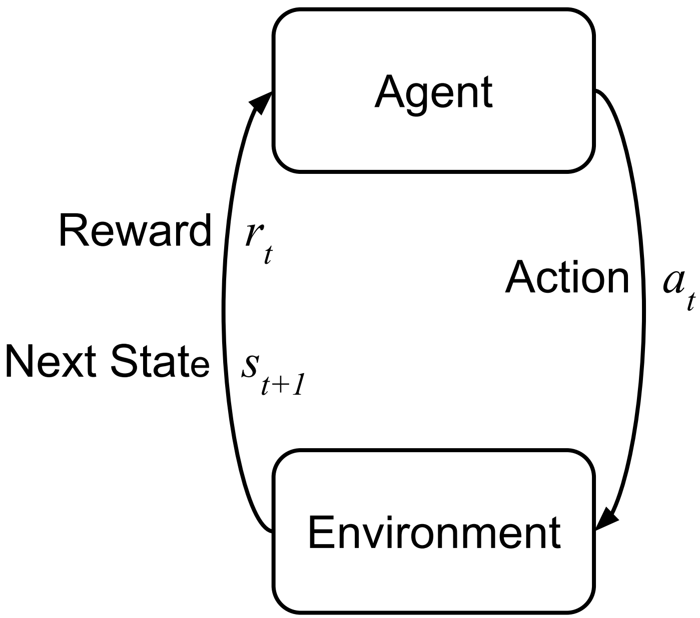
Piyush Khandelwal, Elad Liebman,
Scott Niekum, and Peter Stone

University of Texas at Austin

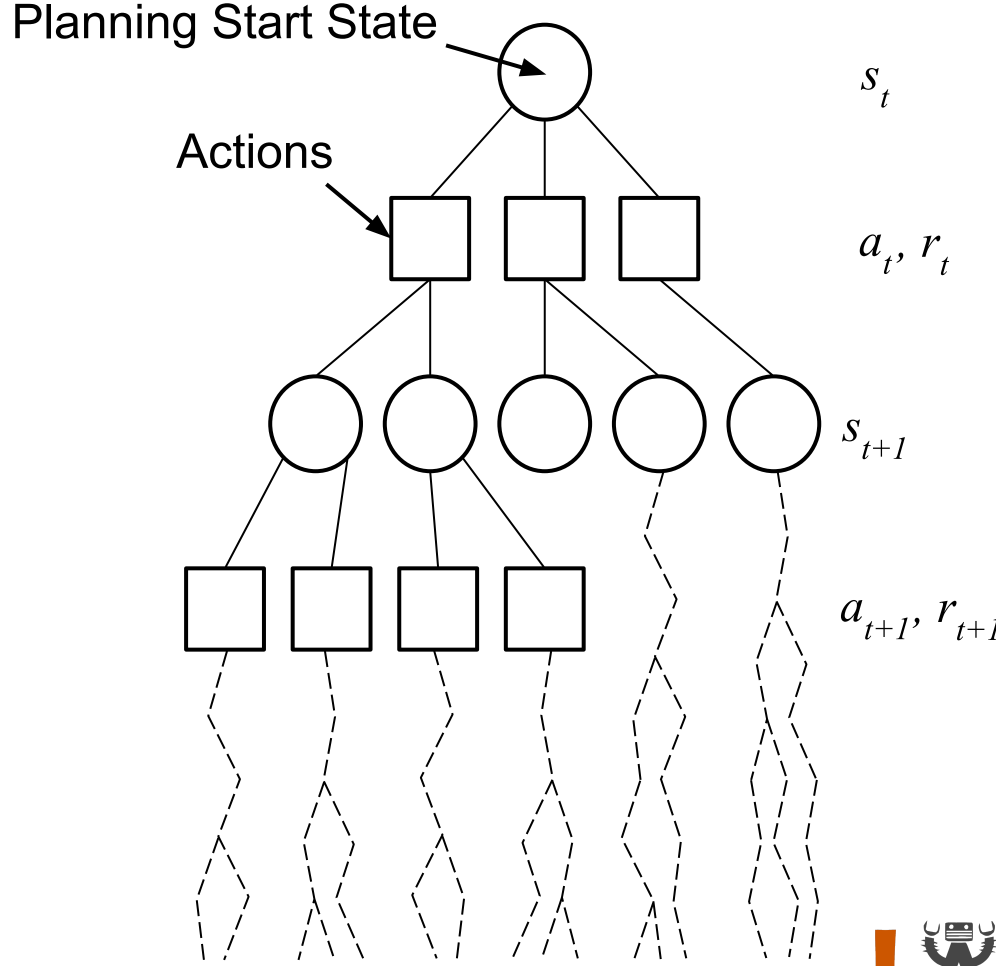
ICML 2016

Monte Carlo Tree Search

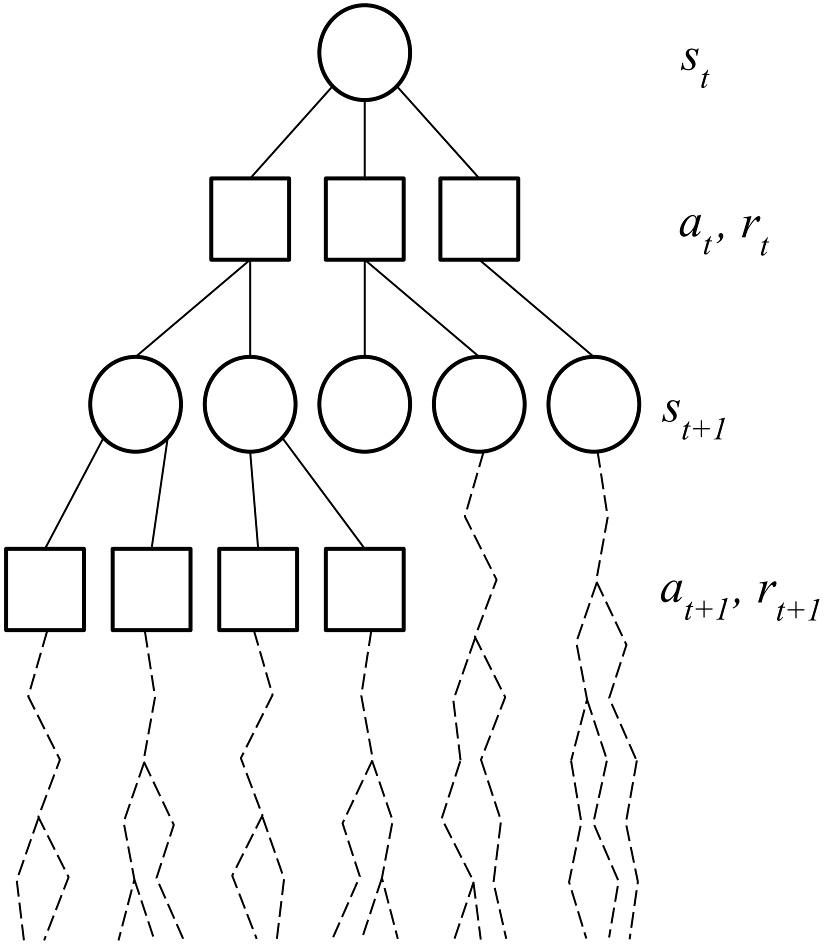
MDP



MCTS



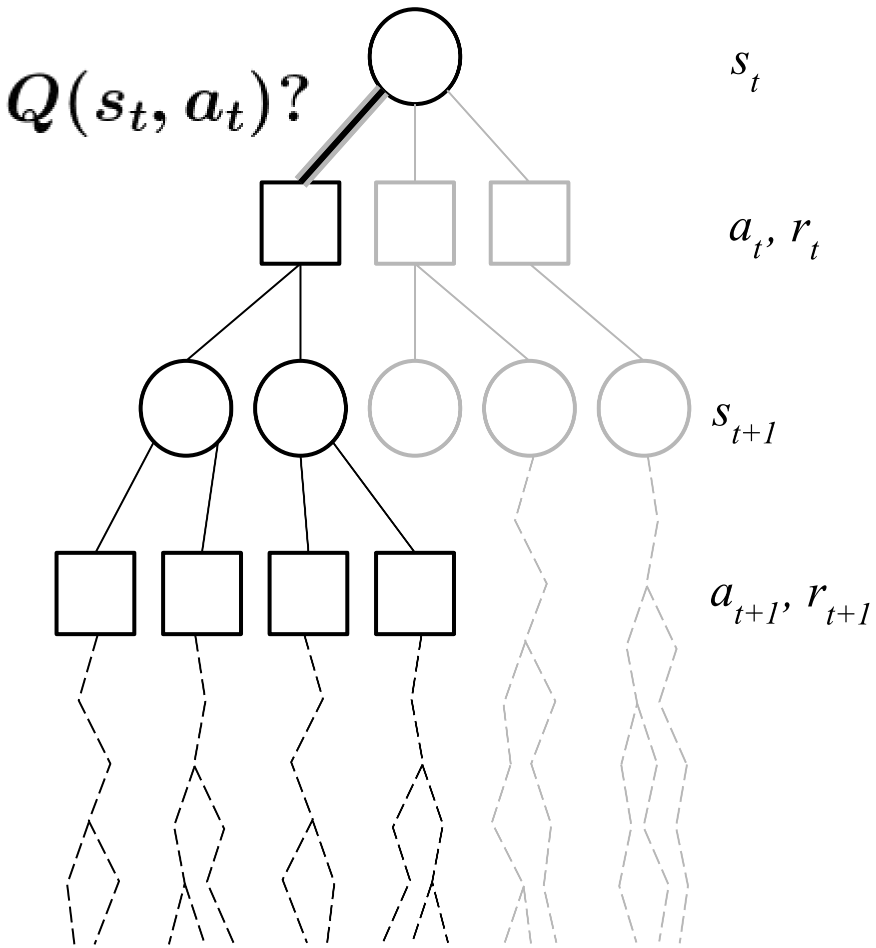
Monte Carlo Tree Search



4 stages in MCTS:

- ➢ Selection
- ➢ Expansion
- ➢ Simulation
- ➢ **Backpropagation**

MCTS - Backpropagation (Motivation)



Monte Carlo backup for single trajectory:

$$R = \sum_{i=0}^{L-1} \gamma^i r_{t+i}$$

Across all trajectories:

$$Q(s_t, a_t) = \mathbb{E}\left[\sum_{i=0}^{L-1} \gamma^i r_{t+i}\right]$$

Can we do better?

This talk

Contribution:

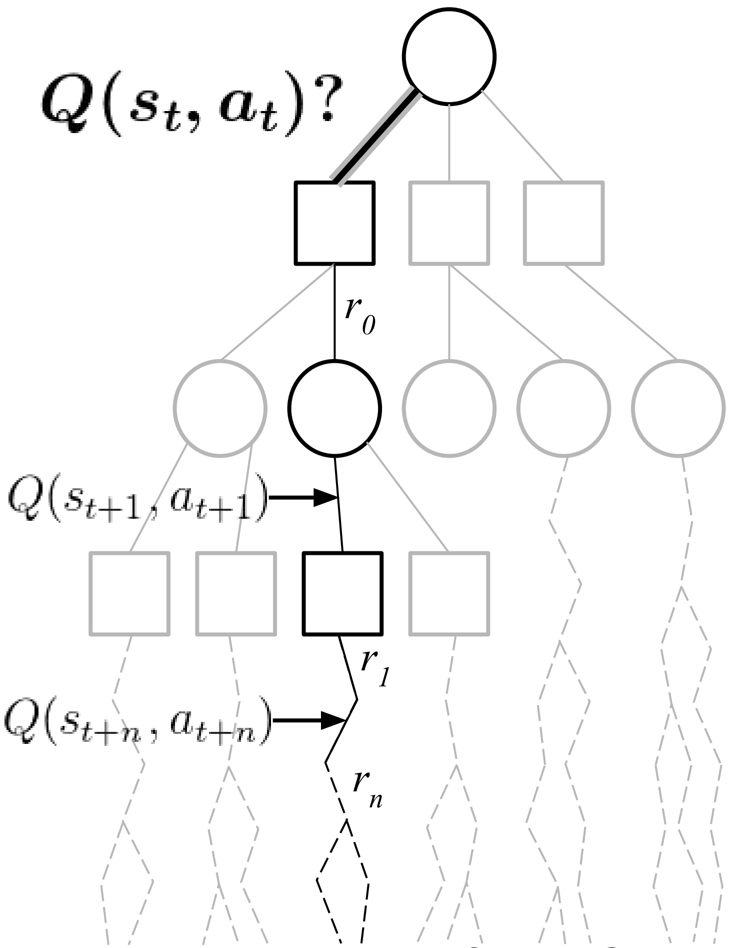
- ➢ Formalize and analyze different on-policy/off-policy complex backup approaches from RL literature for MCTS planning.

Talk outline:

- ➢ Review complex backup strategies from RL in MCTS context.
- ➢ Empirical evaluation using IPC benchmarks.
- ➢ Explore relationship between domain structure and backup strategy performance.

Learning Agents Research Group
The University of Texas at Austin

n-step return (bias-variance tradeoff)



We have estimates for all Q values while performing backpropagation.

We can compute the return sample in many different ways!

1-step:

$$R^{(1)} = r_t + \gamma Q(s_{t+1}, a_{t+1}),$$

n-step:

$$R^{(n)} = \left[\sum_{i=0}^{n-1} \gamma^i r_{t+i}\right] + \gamma^n Q(s_{t+n}, a_{t+n})$$

Monte Carlo:

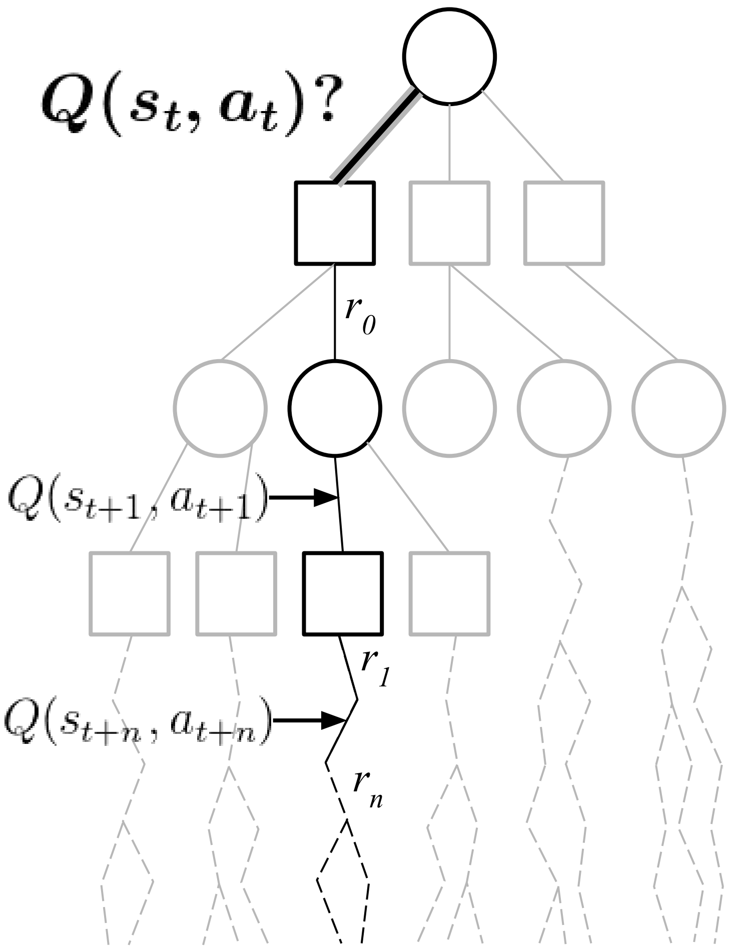
$$R = \sum_{i=0}^{L-1} \gamma^i r_{t+i}$$

More Bias

More Variance

LARG Learning Agents Research Group The University of Texas at Austin

MCTS - Complex return



Complex return: $$R^C = \sum_{i=1}^{L} \left[w_{n,L} \cdot R^{(n)} \right]$$

λ-return/eligibility [Rummery 1995]:

➡ **MCTS(λ)**

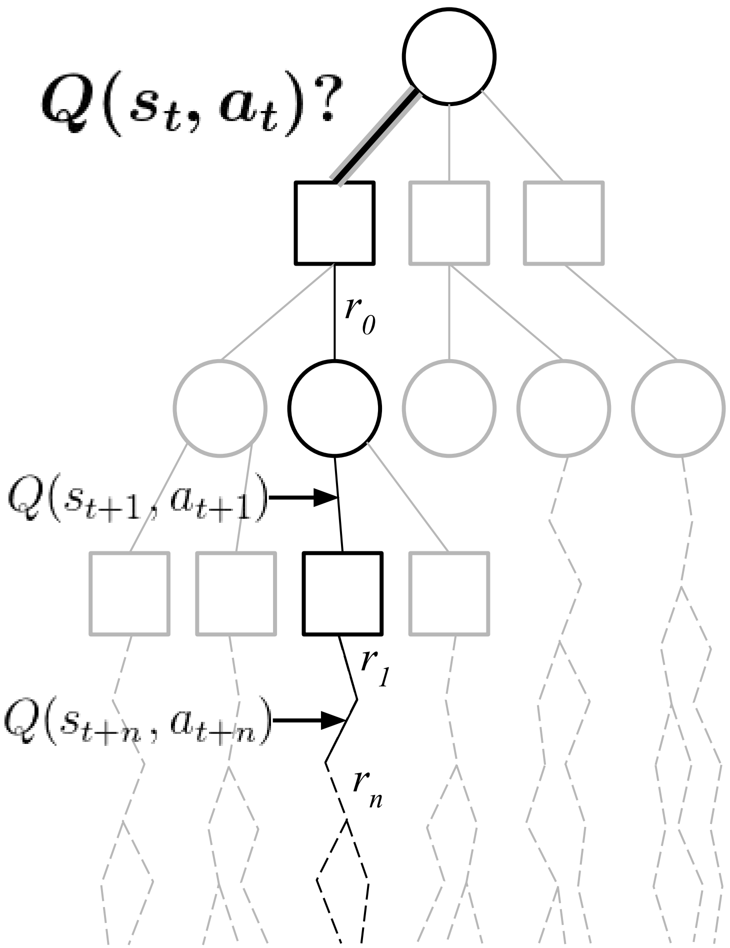
$$w_{n,L}^{\lambda} = \begin{cases} (1-\lambda)\lambda^{n-1} & 1 \leq n < L \\ \lambda^L & n = L \end{cases}$$

γ-return weights [Konidaris et al. 2011]:

➡ **MCTSγ**

$$w_{n,L}^{\gamma} = \frac{\left(\sum_{i=1}^{n} \gamma^{2(i-1)}\right)^{-1}}{\sum_{n=1}^{L} \left(\sum_{i=1}^{n} \gamma^{2(i-1)}\right)^{-1}}$$

MCTS - Complex return



Complex return: $$R^C = \sum_{i=1}^{L} \left[w_{n,L} \cdot R^{(n)} \right]$$

λ-return/eligibility [Rummery 1995]:

➡ MCTS(λ)

$$w^{\lambda}_{n,L} = \begin{cases} (1-\lambda)\lambda^{n-1} & 1 \le n < L \\ \lambda^L & n = L \end{cases}$$

- ➢ Easier to implement.
- ➢ Assumes n-step return variances increase @ λ^{-1}.

γ-return weights [Konidaris et al. 2011]:

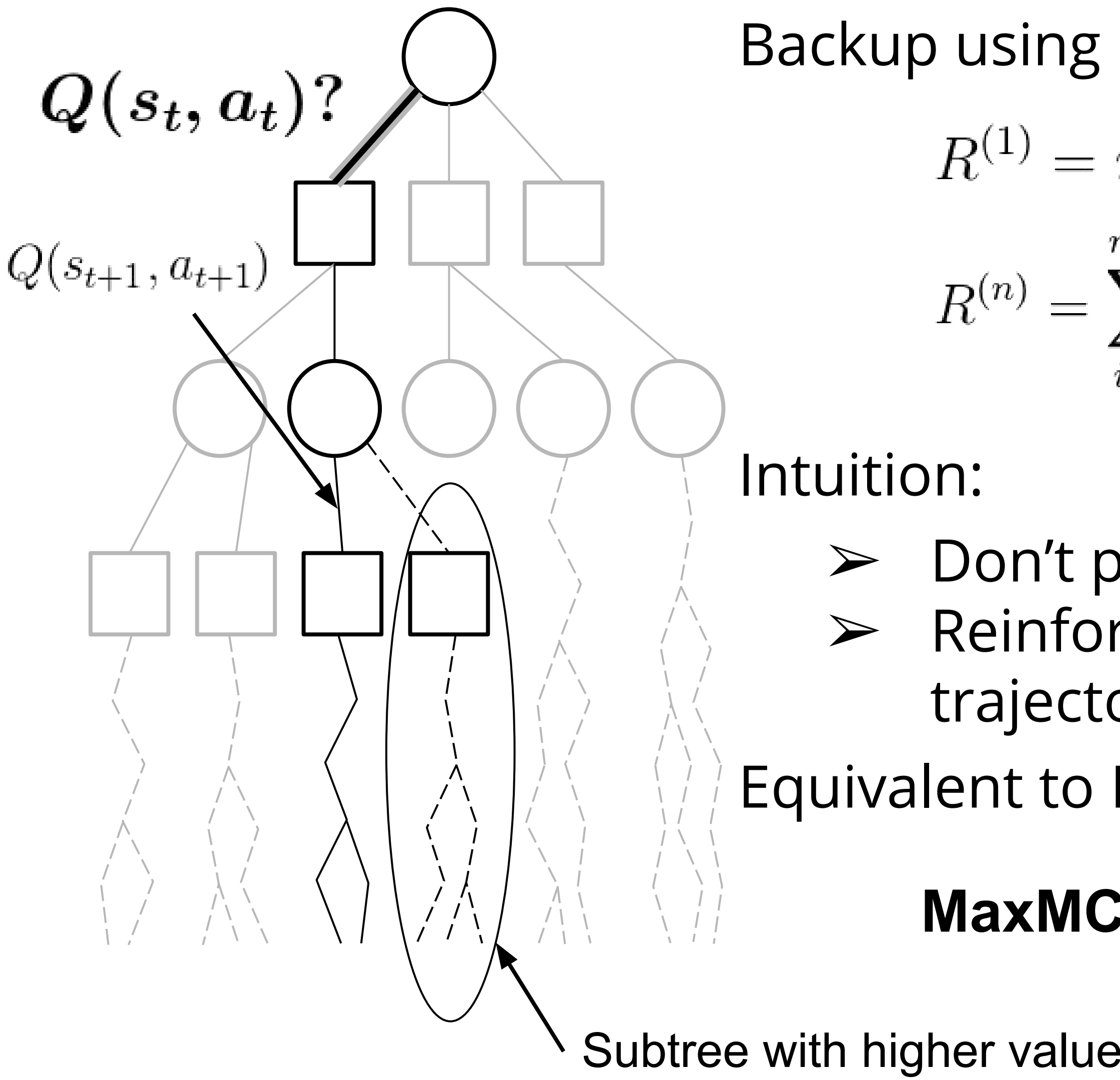
➡ MCTSγ

$$w^{\gamma}_{n,L} = \frac{\left(\sum_{i=1}^{n} \gamma^{2(i-1)}\right)^{-1}}{\sum_{n=1}^{L} \left(\sum_{i=1}^{n} \gamma^{2(i-1)}\right)^{-1}}$$

- ➢ Parameter free.
- ➢ Assumes n-step return variances are highly correlated.

LARG Learning Agents Research Group The University of Texas at Austin

MaxMCTS - Off-policy style returns



Backup using best known action:

$$R^{(1)} = r_t + \gamma \max_a Q(s_{t+1}, a)$$

$$R^{(n)} = \sum_{i=0}^{n-1} \gamma^i r_{t+i} + \gamma^n \max_a Q(s_{t+n}, a)$$

Intuition:

- ➢ Don't penalize exploratory actions.
- ➢ Reinforce previously seen better trajectories instead.

Equivalent to Peng's Q(λ) style updates.

MaxMCTS(λ) and **MaxMCTSγ**

Experiments

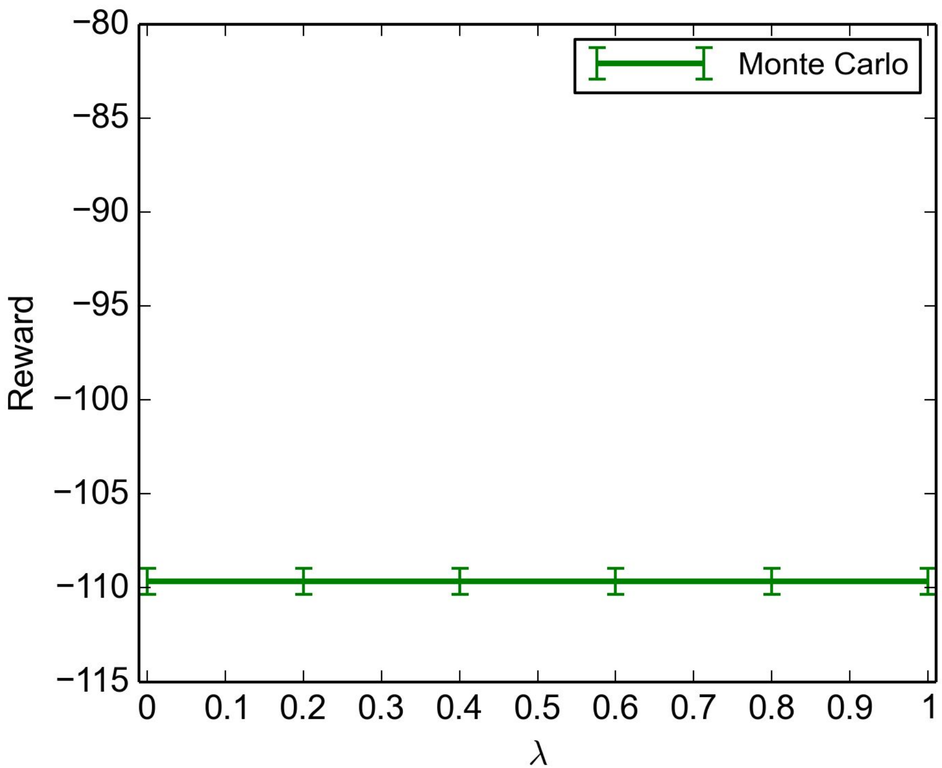
- 4 variants:
  - On-policy: MCTS(λ) and MCTS$_\gamma$
  - Off-policy: MaxMCTS(λ) and MaxMCTS$_\gamma$

- Test performance in IPC domains
  - Limited planning time (10,000 rollouts per step).

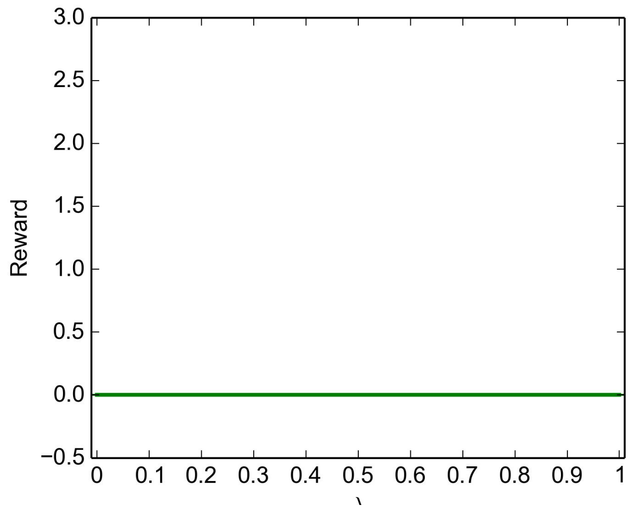
- Grid-world experiments to explore dependency between domain structure and backup strategy performance.

LARG
Learning Agents Research Group
The University of Texas at Austin

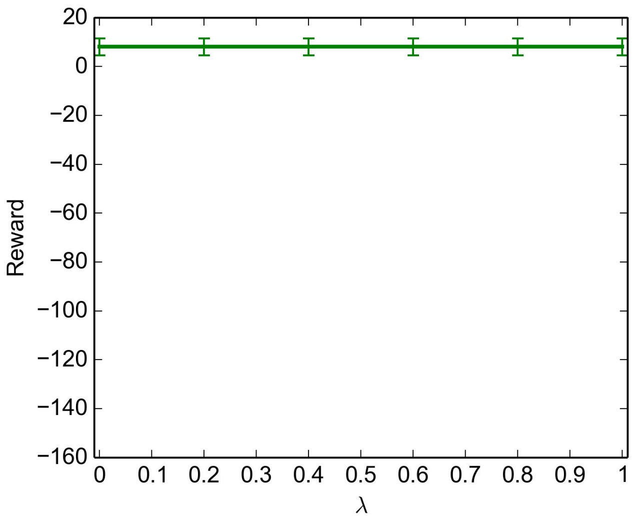
IPC - Random action selection



Elevators

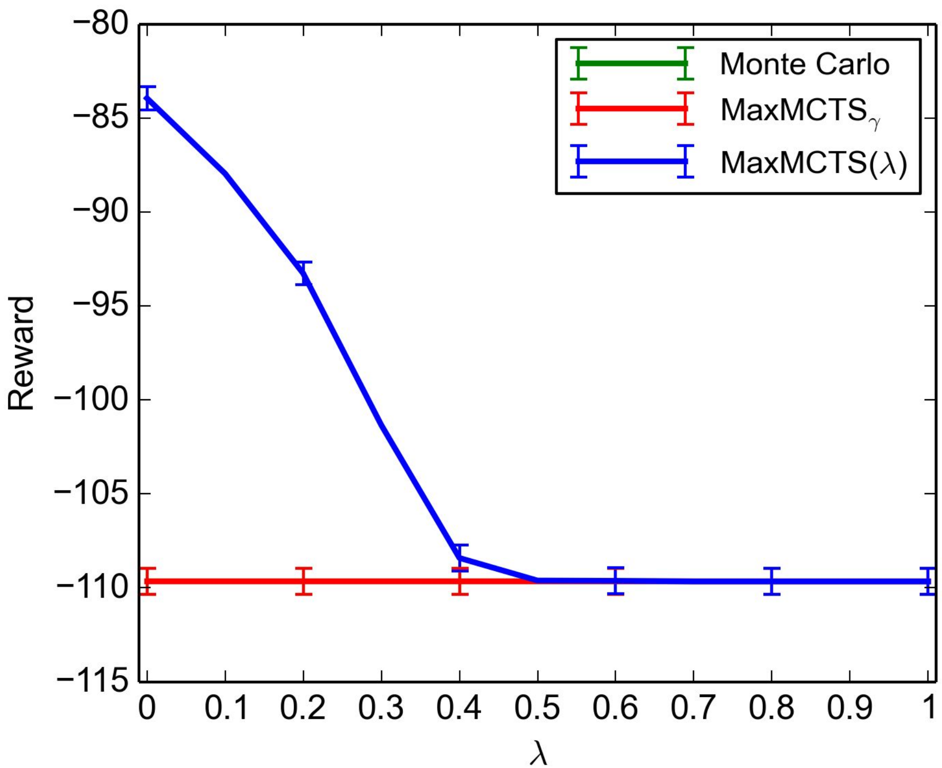


Recon

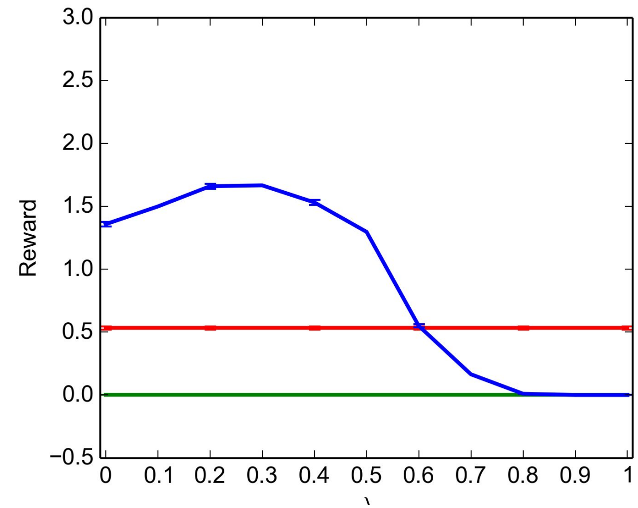


Skill Teaching

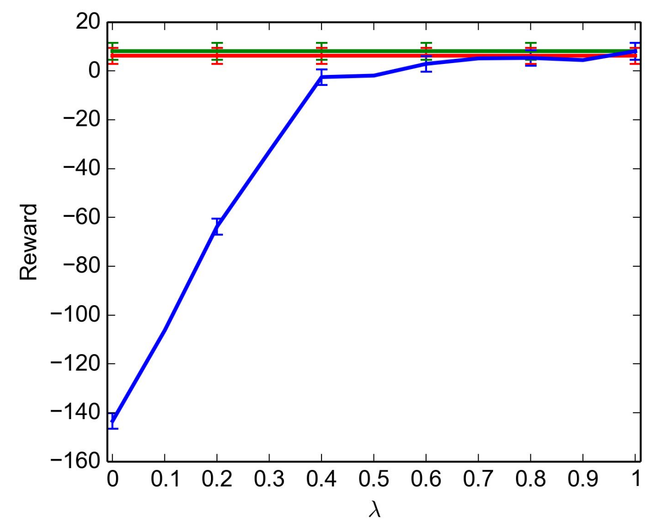
IPC - Random action selection



Elevators

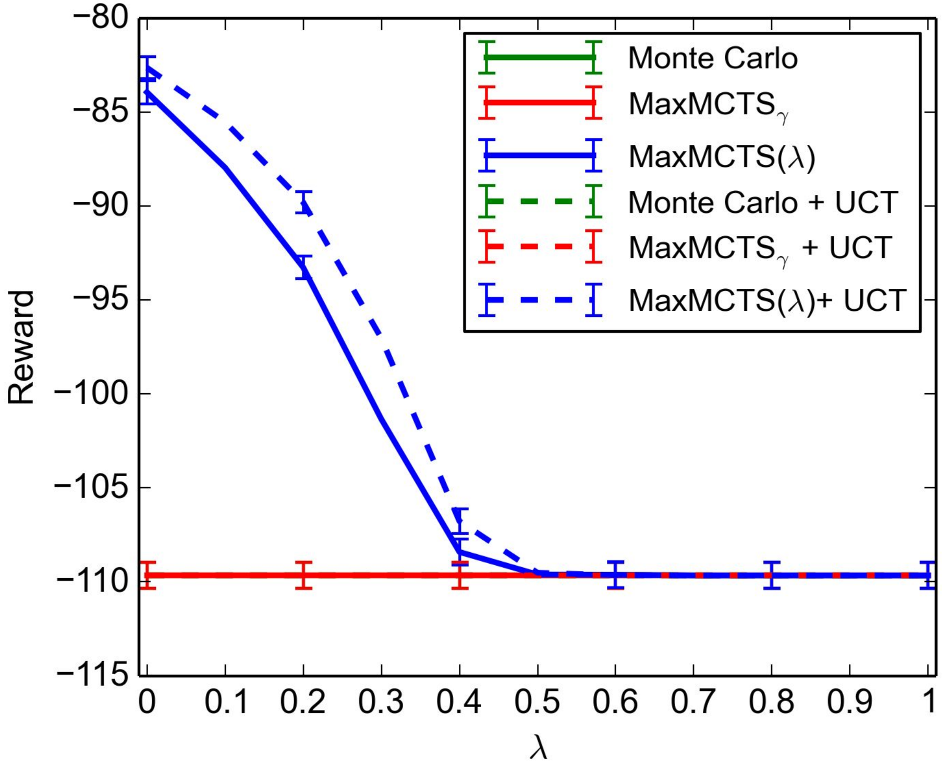


Recon

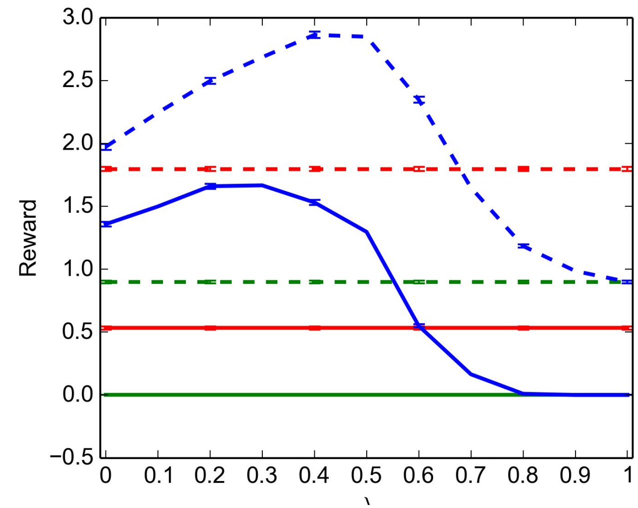


Skill Teaching

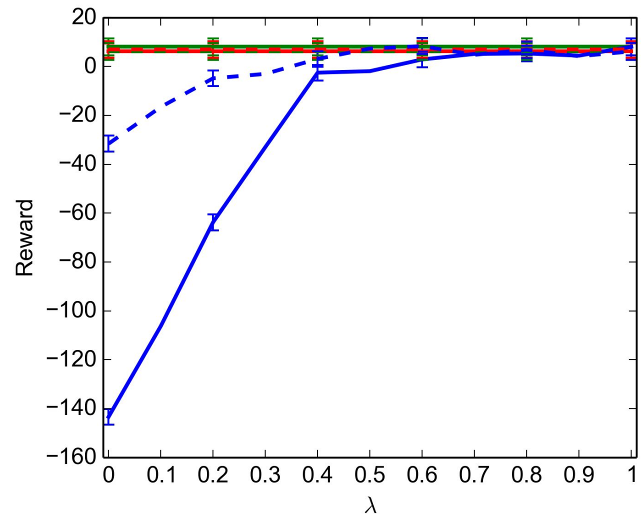
IPC - UCB1 action selection



Elevators

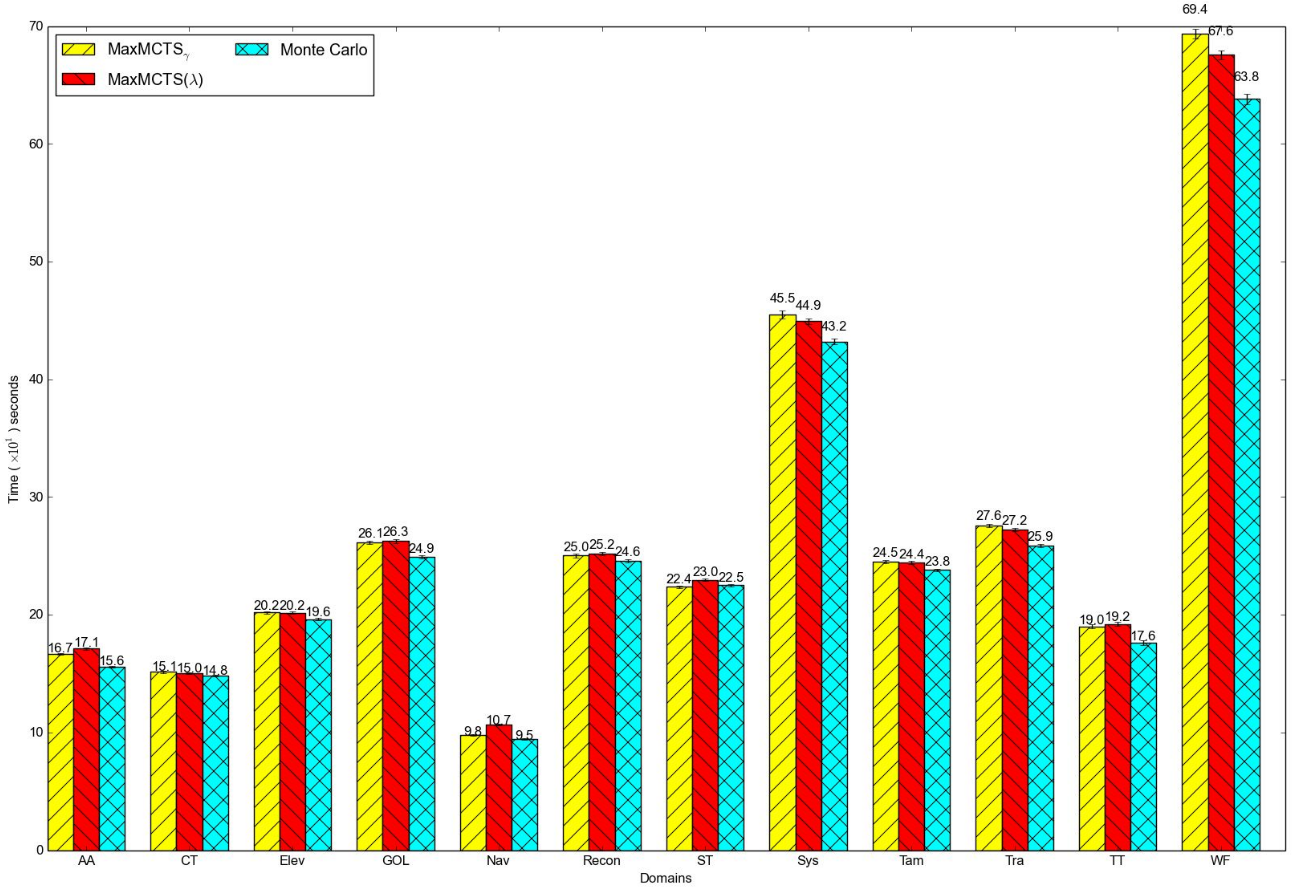


Recon

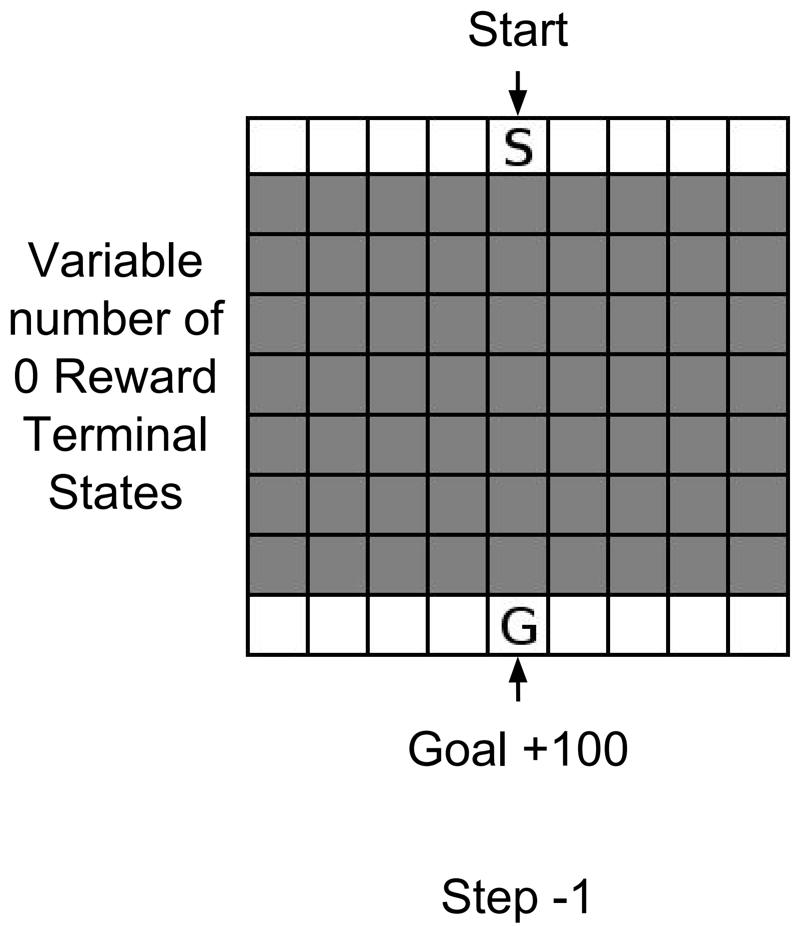


Skill Teaching

Computational Time Comparison



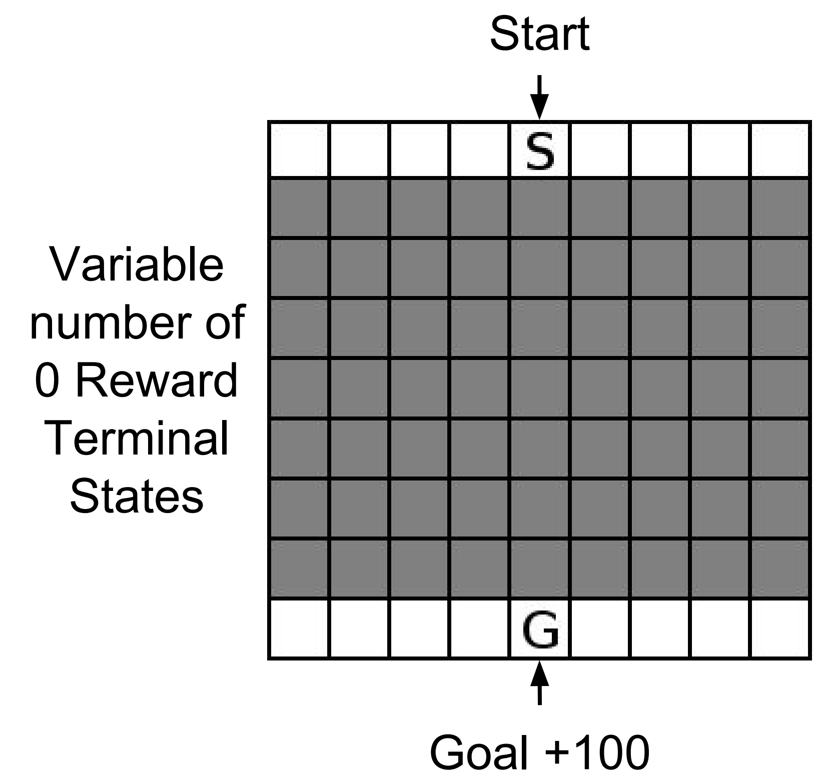
Grid World Domain



- 90% chance of moving in intended direction.
- 10% chance of moving to any neighbor randomly.

Learning Agents Research Group
The University of Texas at Austin

Grid World Domain



Step -1

#0-Term	0	3	6	15
$\lambda = 1$	***90.4***	11.3	0.9	-2.2
$\lambda = 0.8$	90.2	28.0	10.7	-1.4
$\lambda = 0.6$	89.5	62.8	45.3	8.5
$\lambda = 0.4$	88.7	***85.1***	77.6	24.1
$\lambda = 0.2$	87.7	82.6	**78.1**	28.4
$\lambda = 0$	84.5	79.8	74.1	**31.8**

Learning Agents Research Group
The University of Texas at Austin

Related Work

- λ-return has been applied previously for planning:
  - TEXPLORE used a slightly different version of MaxMCTS(λ) [Hester 2012].
  - Dyna2 used eligibility traces [Silver et al. 2008].

- Other backpropagation strategies:
  - MaxMCTS(λ=0) is equivalent to MaxUCT [Keller, Helmert 2012].
  - Coulom analyzed hand-designed backpropagation strategies in 9x9 Computer Go [Coulom 2007].

- Planning Horizon:
  - Dependence of planning horizon on performance [Jiang et al. 2015].

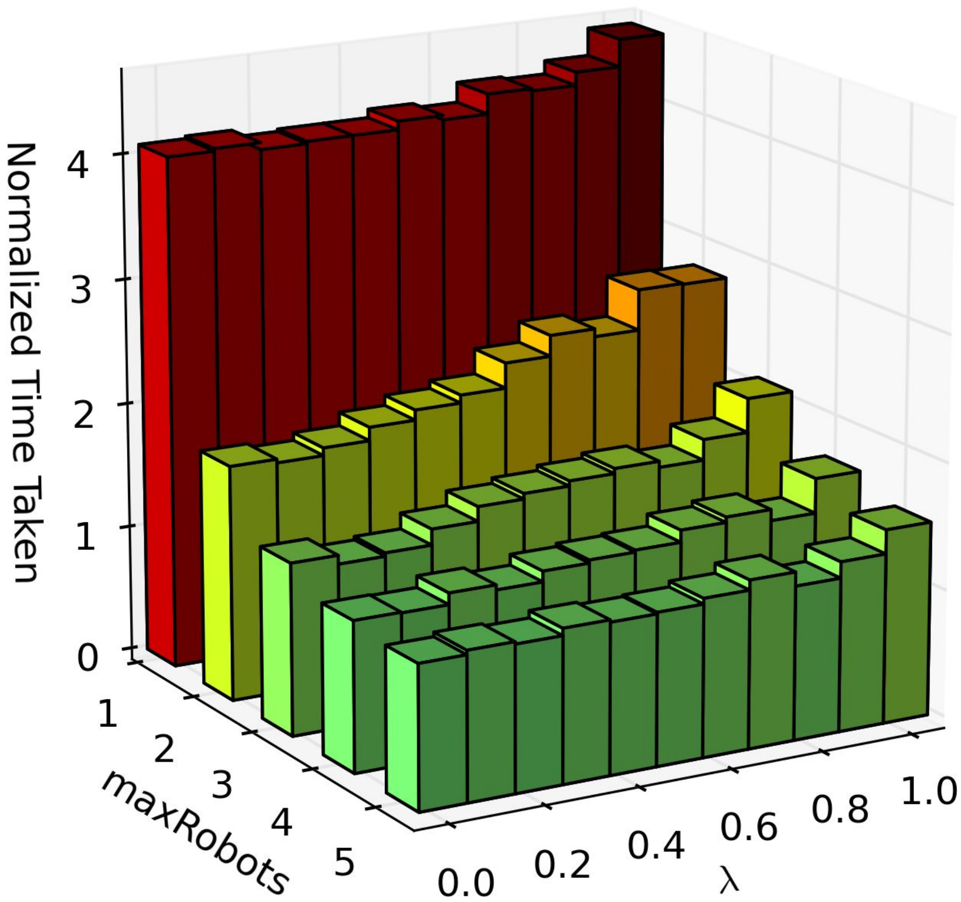
Learning Agents Research Group
The University of Texas at Austin

Conclusions

- In some domains, selecting the right complex backup strategy is important.
- MaxMCTSγ is a parameter-free approach that always performs better than/equivalent to Monte Carlo.
- MaxMCTS(λ) performs best if λ can be selected appropriately.
- Backup strategy performance related to number of trajectories with high rewards.

Multi-robot coordination

[Khandelwal et al. 2015]



- 84 discrete and continuous factors
- 100-500 actions per state (10-50 after heuristic reduction).

Learning Agents Research Group
The University of Texas at Austin