Monte-Carlo Tree Search in Verification of Markov Decision Processes

LiVe 2018: 2nd Workshop on Learning in Verification

Pranav Ashok[1], Tomáš Brázdil[2], Jan Křetínský[1] and Ondřej Slámečka[2]

April 20, 2018

[1]Technical University of Munich, Germany

[2]Masaryk University, Brno, Czech Republic

Motivation

Problem

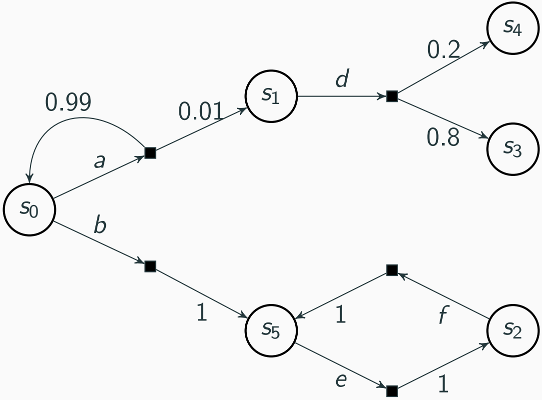
Computing the maximum probability of reaching a set of goal states in a Markov Decision Process

- Traditional approaches: Value/Strategy Iteration and LP
- New approaches inspired by learning (BRTDP etc.)
- Successful AI techniques such as Monte-Carlo Tree Search

Contributions

Develop a spectrum of algorithms based on MCTS and pit them against each other

Markov Decision Process (MDPs)



Reachability Objective

Probability of reaching a set of goal states under the best possible choice of actions

$$P_{max} = \sup_{\sigma \in \mathcal{S}} \mathbb{P}^{\sigma}[\Diamond G]$$

Value Iteration

Bellman Equations

$$V(s) = \begin{cases} 1 & \text{if } s = \mathtt{1} \\ 0 & \text{if } s = \mathtt{0} \\ \max_a \sum_{s' \in S} P(s, a, s') \cdot V(s') & \text{otherwise} \end{cases}$$

Solving the above iteratively is known as **Value Iteration**

- $V_0(s) = 0$ for all $s \neq \mathtt{1}$
- V_n computed from V_{n-1}

Bounded Value Iteration[1]

Standard VI: Convergence in limit, but no stopping criterion

An easy remedy for this is to run two value iterations

1. VI initialized with $V_0(s) = 0 \quad$ (except $V_0(\imath) = 1$)
2. VI initialized with $V_0(s) = 1 \quad$ (except $V_0(\mathfrak{o}) = 0$)

[1] Idea explored by McMahan et. al. 2005 and Haddad et. al. 2014

Asynchronous VI

Convergence Theorem

To ensure convergence, every state must be updated infinitely often

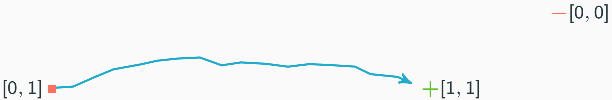
$\Longrightarrow$ lot of freedom in choosing the order of updates

Bounded Real-time Dynamic Programming (BRTDP) [2,3]

- Used to compute ε-optimal strategy (and value)
- Every state has a L/U bound on probability of reaching goal
- Repeatedly samples paths from initial state
- Back-propagates values along the path using VI operator

[2] McMahan et. al., Bounded Real-Time Dynamic Programming, ICML '05
[3] Brazdil et. al., Verification of Markov Decision Processes using Learning Algorithms, ATVA '14

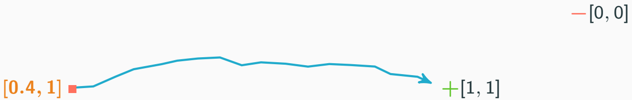
BRTDP in action



Summary

1. Next choice based on greatest U bound
2. Resolve probabilities randomly
3. Back-propagate info using bellman updates

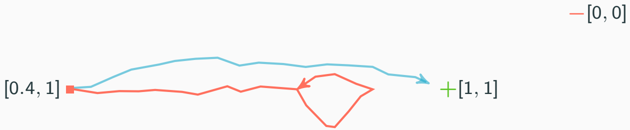
BRTDP in action



Summary

1. Next choice based on greatest U bound
2. Resolve probabilities randomly
3. Back-propagate info using bellman updates

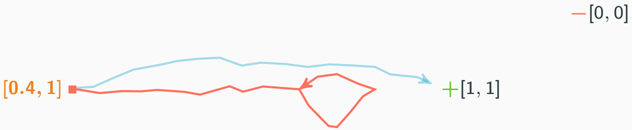
BRTDP in action



Summary

1. Next choice based on greatest U bound
2. Resolve probabilities randomly
3. Back-propagate info using bellman updates

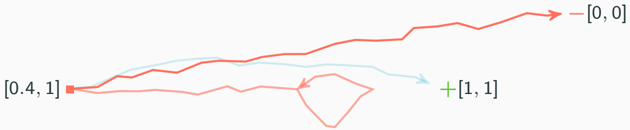
BRTDP in action



Summary

1. Next choice based on greatest U bound
2. Resolve probabilities randomly
3. Back-propagate info using bellman updates

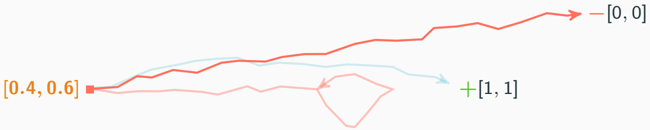
BRTDP in action



Summary

1. Next choice based on greatest U bound
2. Resolve probabilities randomly
3. Back-propagate info using bellman updates

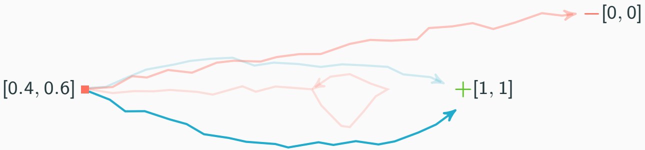
BRTDP in action



Summary

1. Next choice based on greatest U bound
2. Resolve probabilities randomly
3. Back-propagate info using bellman updates

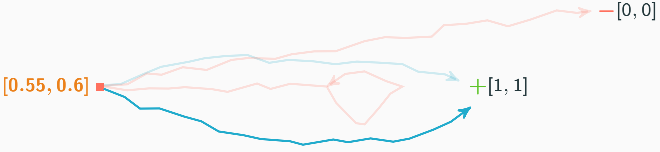
BRTDP in action



Summary

1. Next choice based on greatest U bound
2. Resolve probabilities randomly
3. Back-propagate info using bellman updates

BRTDP in action



Summary

1. Next choice based on greatest U bound
2. Resolve probabilities randomly
3. Back-propagate info using bellman updates

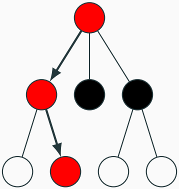
Monte-Carlo Tree Search

Timeline

1987	Idea explored by Bruce Abramson in his PhD Thesis
1992	Employed in a Go program by Bernd Brügmann
2006	Rémi Coulom coins the term Monte-Carlo Tree Search
2008	MoGo starts winning against strong amateur players
2016	AlphaGo beats Lee Sedol

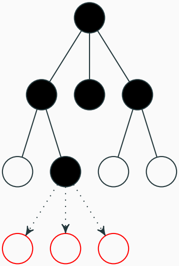
Monte-Carlo Tree Search

Select using **tree policy**

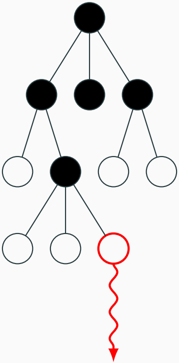


Monte-Carlo Tree Search

Expand and add one or more children

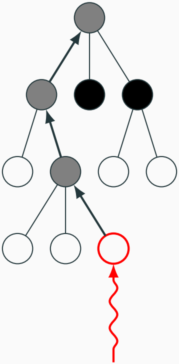


Roll-out simulations using **roll-out policy**

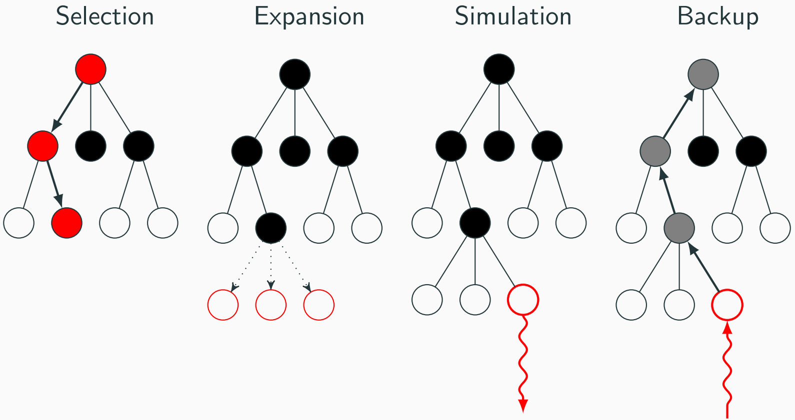


Monte-Carlo Tree Search

Backup values to the root



MCTS: The four stages



Instantiation: Tree Policy

Tree Policy

Picks the state from which roll-out/simulation is run

Upper Confidence Bound

proportion of successful rollouts

$$UCB(t) = \frac{v_t}{n_t} + C\sqrt{\frac{\ln n_{\text{parent}(t)}}{n_t}}$$

exploration encouraging term

Pick successor with greatest UCB value

Instantiation: Roll-out Policy

Option 1: Less-informed

Random roll-outs: uniform policy

Option 2: More-informed

BRTDP style: pick action with greatest upper bound

Instantiation: Roll-out Policy

Option 1: Less-informed
Random roll-outs: uniform policy

Option 2: More-informed
BRTDP style: pick action with greatest upper bound

For sake of guarantees, maintain and backup L/U bounds
for all states on roll-out

Spectrum of MCTS based algorithms

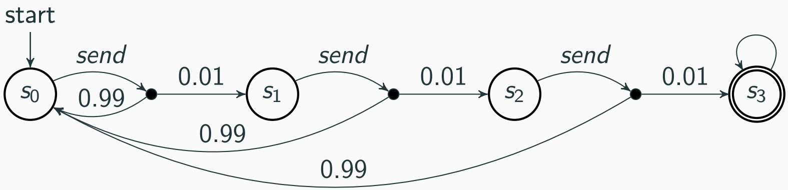
Spectrum of algorithms ranging from pure MCTS to pure BRTDP

	MCTS	BMCTS	MCTS-BRTDP	BRTDP-UCB	BRTDP
Tree Policy	UCB	UCB	UCB	UCB	U
Roll-out Policy	Uniform	Uniform	BRTDP		
Roll-out Backup	—	L, U	L, U	L, U	L, U
Tree Backup	—	L, U	L, U		

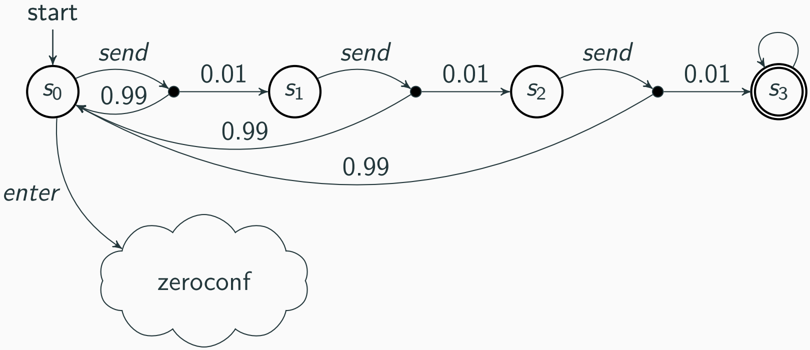
Experiments

Benchmark	BMCTS	MCTS-BRTDP	BRTDP-UCB	BRTDP	VI
consensus	5.55	6.48	7.47	6.15	1.13
leader	18.67	15.79	16.33	15.06	8.94
mer	—	4.79	—	3.63	—
firewire	0.07	0.08	0.09	0.09	6.99
wlan	0.09	0.07	0.08	0.08	—
zeroconf	0.93	0.20	0.59	0.20	—
comp-firewire	9.36	9.55	—	—	20.77
comp-wlan	2.51	2.25	—	—	—
comp-zeroconf	—	29.55	—	—	—
branch-firewire	0.09	0.09	0.02	0.09	9.33
branch-wlan	0.10	0.08	0.09	0.07	—
branch-zeroconf	25.90	30.78	35.67	38.14	—

Special Models

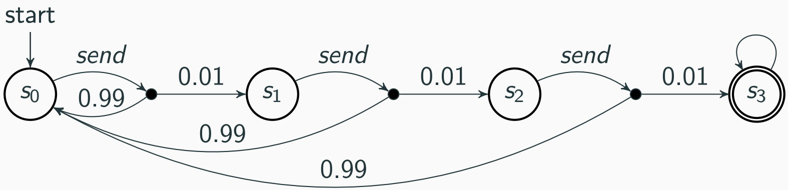


Special Models



branch-zeroconf

Special Models



||

composition-zeroconf

Summary

Observations about MCTS-based techniques

1. Can handle rare events better (if tree encompasses it)
2. Performs good when degree of non-determinism is low
3. Sometimes the tree overhead does not justify ditching BRTDP

Summary

Observations about MCTS-based techniques

1. Can handle rare events better (if tree encompasses it)
2. Performs good when degree of non-determinism is low
3. Sometimes the tree overhead does not justify ditching BRTDP

Take-away message

1. Exploration + exploitation $\implies$ more problems solvable (in reasonable time)
2. Benefits from starting sampling away from the initial state

Summary

Observations about MCTS-based techniques

1. Can handle rare events better (if tree encompasses it)
2. Performs good when degree of non-determinism is low
3. Sometimes the tree overhead does not justify ditching BRTDP

Take-away message

1. Exploration + exploitation $\implies$ more problems solvable (in reasonable time)
2. Benefits from starting sampling away from the initial state

Future Work

1. MCTS with non-UCB policies
2. Sampling from non-initial states (without MCTS)