Bayesian Mixture Modeling and Inference based Thompson Sampling in Monte-Carlo Tree Search

Aijun Bai

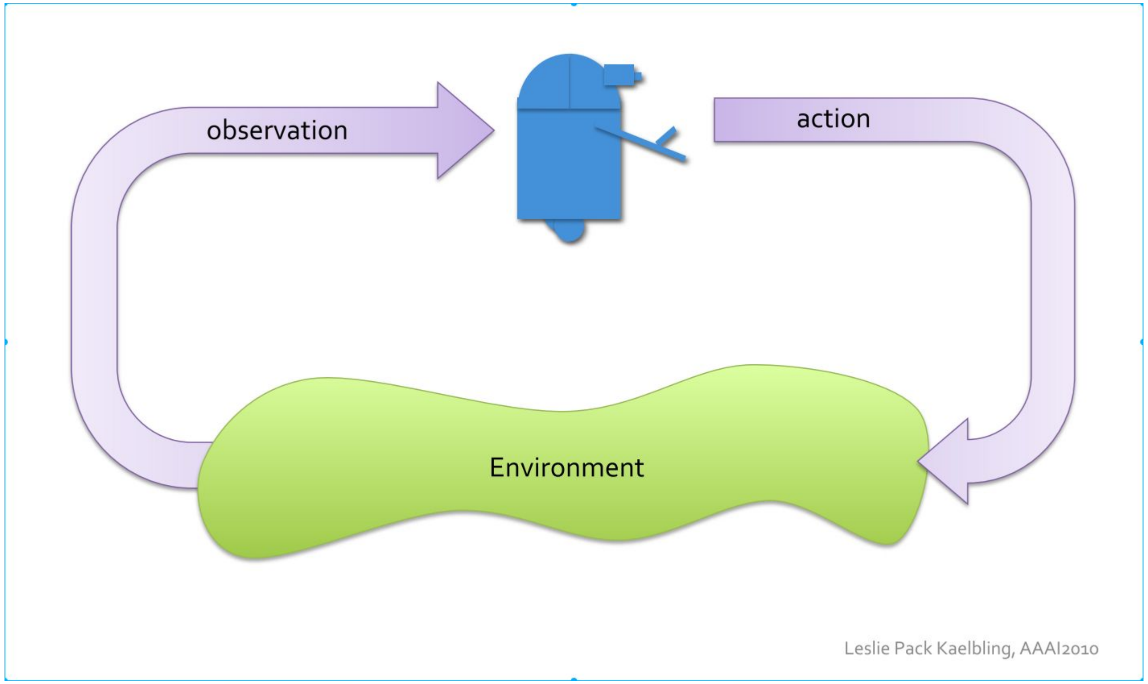
Oct. 11, 2013

Background

- Planning under Uncertainty
- Markov Decision Processes
- AND/OR Graph for MDPs
- Monte Carlo Tree Search
- Multi-Armed Bandits
- Upper Confidence Bound Heuristic
- UCB applied to Trees
- Bayesian Modeling and Inference
- Thompson Sampling

Planning under Uncertainty

- Sequential decision-making
- Action uncertainty
- Observation uncertainty

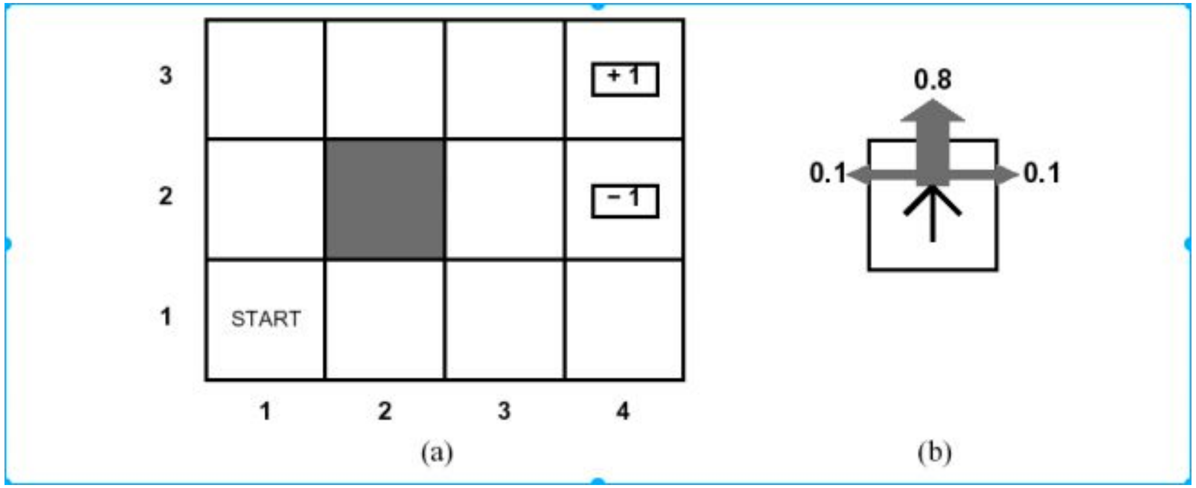


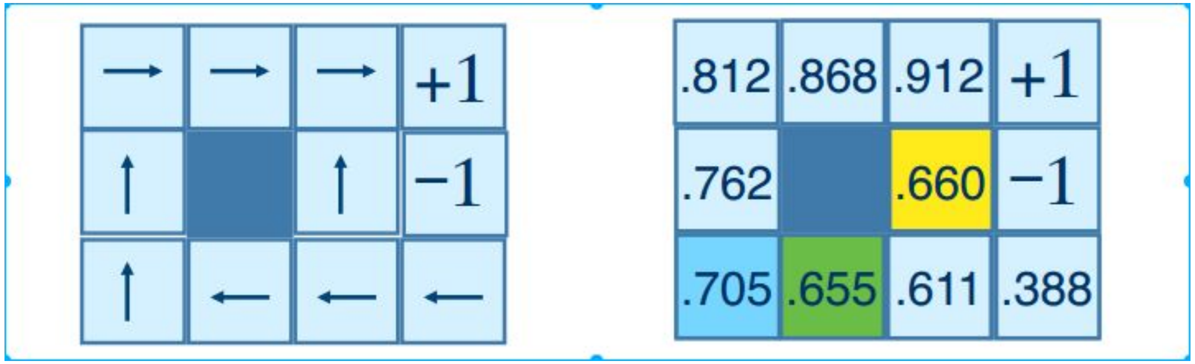
Markov Decision Processes

- Fully observable
- An MDP is a 4-tuple
  - State space: S
  - Action space: A
  - Transition function: $T(s' \mid s, a)$
  - Reward function: $R(s, a)$
- Policy
  - $\pi(s)$: S -> A
  - Optimal policy

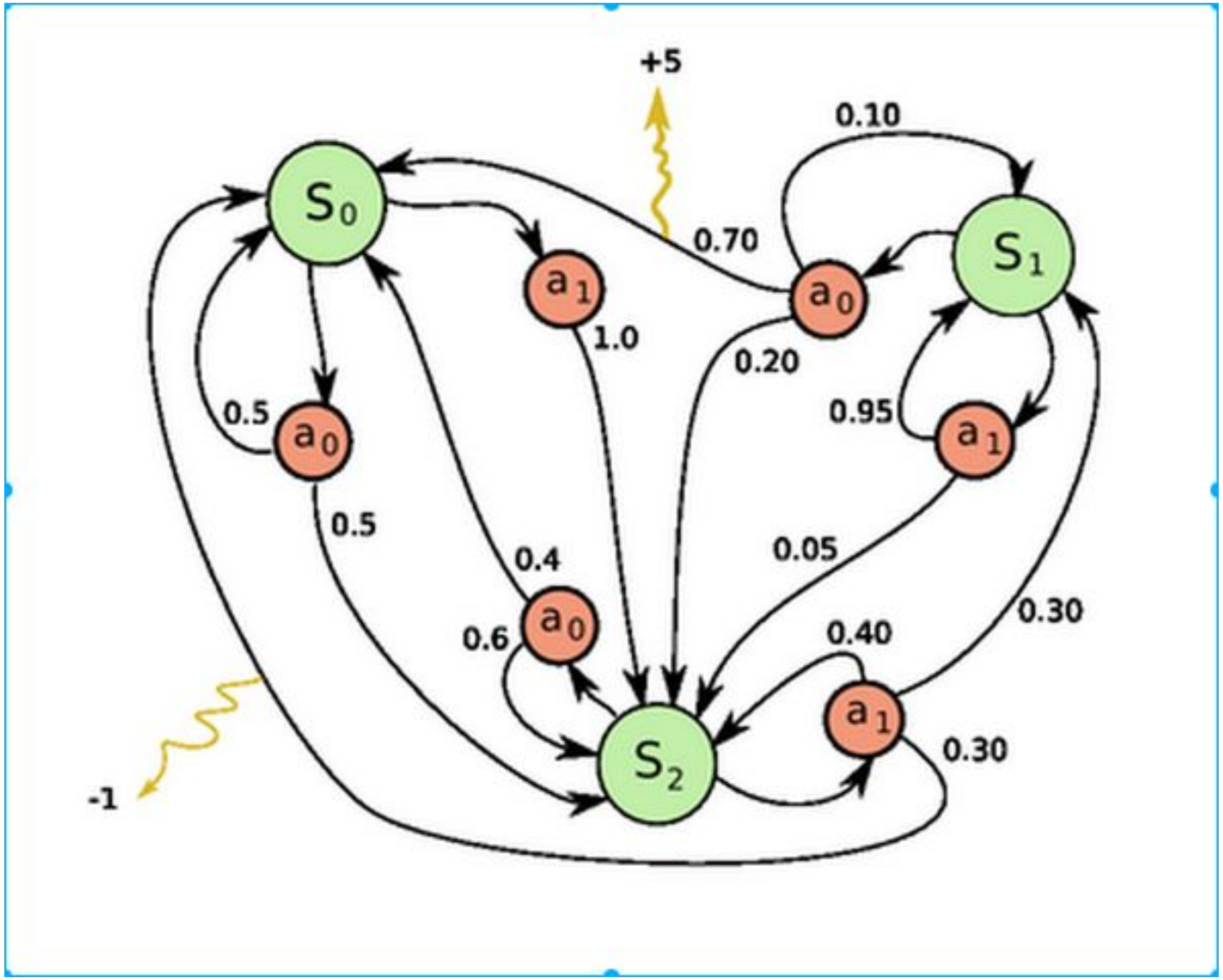
- Algorithms
  - Offline
    - Linear Programming
    - Value Iteration
    - Policy Iteration
  - Online
    - AND/OR Graph Search
    - Real Time Dynamic Programming
    - Monte-Carlo Tree Search

Example: A Grid Environment

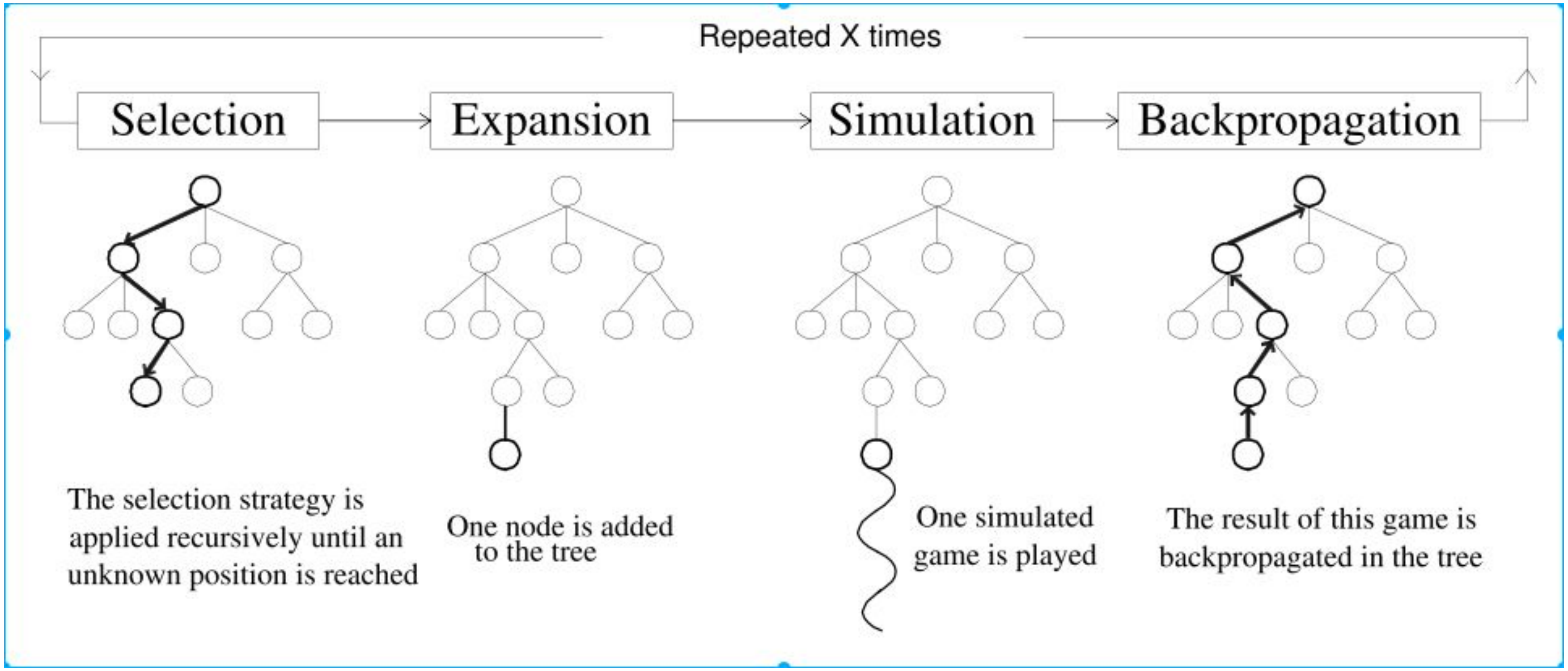




AND/OR Graph for MDPs



Monte-Carlo Tree Search

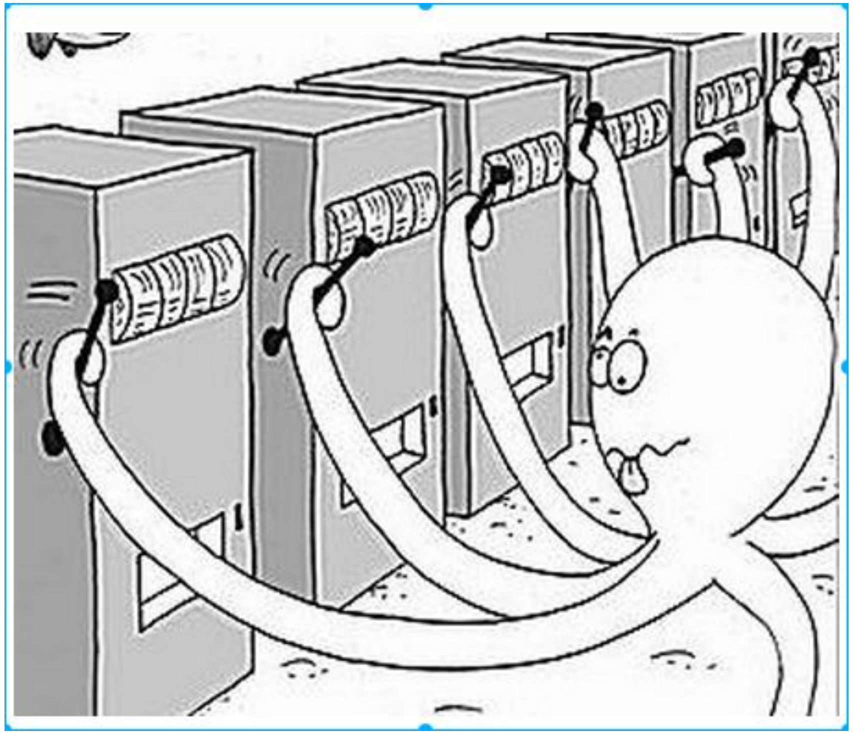


Monte-Carlo Tree Search (cont.)

- Advantages of MCTS
  - Highly selective best-first search
  - Works for black-box simulators
  - No need for explicit mathematical model (*T* or *R*)
  - Computationally efficient, anytime, parallelisable
  - Easily integrated with domain knowledge
    - Nodes initialization: Heuristics
    - Tree policy: Preferred actions
    - Rollout policy: Manually specified
- Applications
  - Game playing, MDP, POMDP, Bayes RL

Multi-Armed Bandits

- MAB
  - *N* slot machines
  - Unknown reward distributions
  - Policy
    - History -> Action
  - Optimal policy
    - Minimizes the cumulative regret
- Example
  - A, 1, B, 0, A, 0, ?



Upper Confidence Bound Heuristic

- UCB heuristic
  - Maximize

$$\hat{r}_i + \sqrt{\frac{2 \log t}{n_i}},$$

  - Convergence
- UCB applied to trees (UCT)
  - Treat each decision node in MCTS as an MAB
  - Maximize

$$\bar{Q}(s,a) + c\sqrt{\log N(s)/N(s,a)}$$

  - Convergence

Bayesian Modeling and Inference

$$p(\theta|\mathbf{X},\alpha) = \frac{p(\mathbf{X}|\theta)p(\theta|\alpha)}{p(\mathbf{X}|\alpha)} \propto p(\mathbf{X}|\theta)p(\theta|\alpha)$$

- θ, hidden parameter
- α, hyper-parameter of θ
- **X**, observed data points
- $p(\theta|\alpha)$, prior distribution
- $p(\boldsymbol{X}|\theta)$, likelihood
- $p(\mathbf{X}|\alpha)$, marginal likelihood
- $p(\theta|\boldsymbol{X}, \alpha)$, posterior distribution
- Conjugate priors => closed form

Thompson Sampling

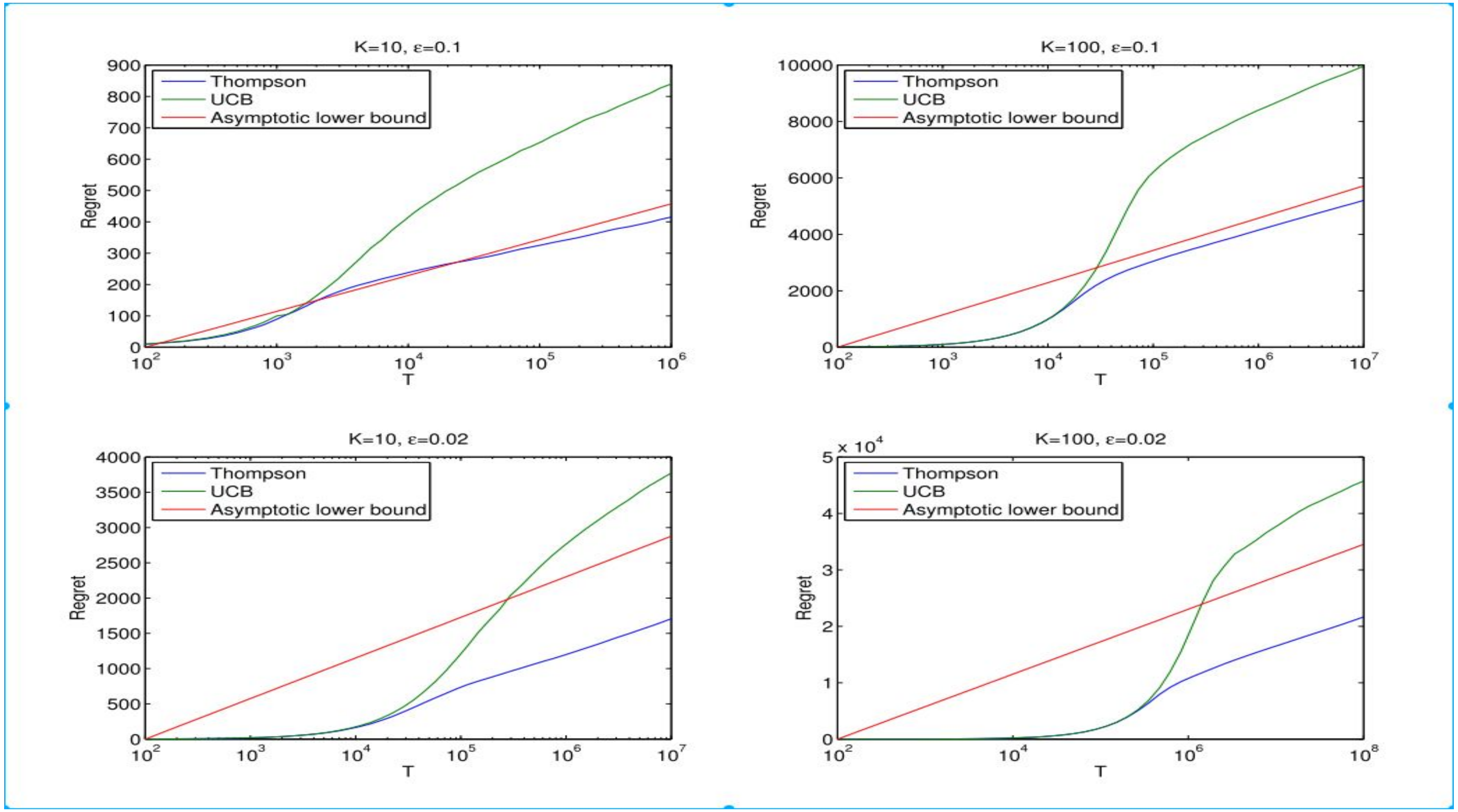
- Probability matching strategies
  - Select actions based on the probabilities of being optimal

$$P(a) = \int \mathbb{1}[a = \operatorname*{argmax}_{a'} E[X_{a'}|\theta_{a'}]] \prod_{a'} P_{a'}(\theta_{a'}|Z)\, \mathrm{d}\boldsymbol{\theta},$$

  - Example: 2 actions *a* and *b*, *P*(*a*) = 0.3, *P*(*b*) = 0.7
  - Efficiently approached by sampling methods
    - Maintain posteriors of reward distributions for each action
      - Updated by Bayesian rules
    - Sample reward distributions according to posteriors
    - Select action with highest expectation $a^* = \operatorname{argmax}_a E[X_a|\theta_a]$
- TS applied to MABs
  - Converges faster and more robust

Thompson Sampling (cont.)

- TS v.s. UCB in MABs



Thompson Sampling (cont.)

- Example:

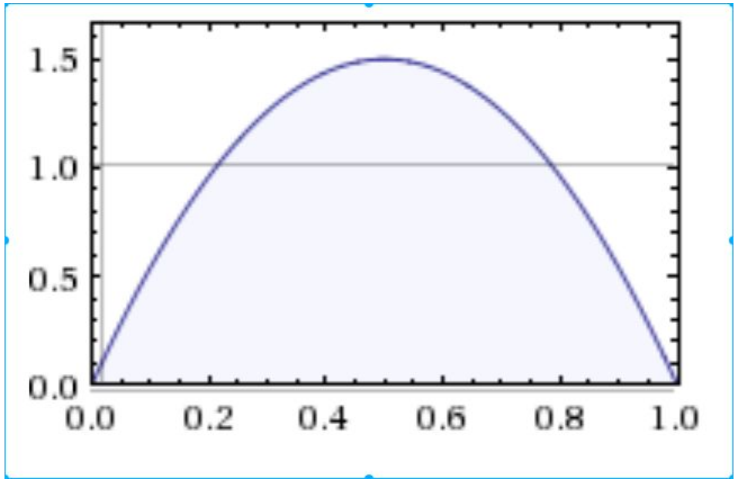
  A, 1, B, 0, A, 0, ?

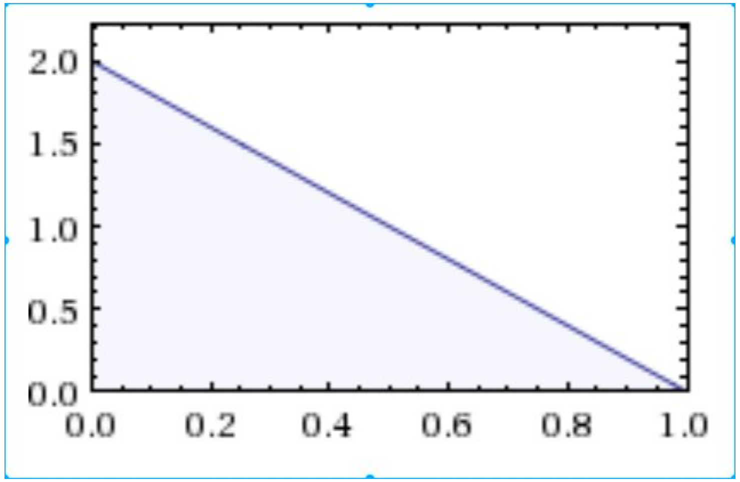
  – Bernoulli p_a and p_b

  – Uniform priors
    - $p_a \sim Uniform(0, 1)$
    - $p_b \sim Uniform(0, 1)$

  – Beta posteriors
    - $p_a \sim Beta(2, 2)$
    - $p_b \sim Beta(1, 2)$

  – Thompson sampling





Dirichlet-NormalGamma MCTS

- DNG-MCTS
  - Bayesian Mixture Modeling and Inference
  - Thompson Sampling
  - Monte-Carlo Tree Search
- Assumptions (according to CLT for MCs)
  - $X_{s,\pi}$ => Normal distribution
  - $X_{s,a,\pi}$ => mixture of Normal distributions
- Bayesian modeling and inference
  - $X_{s,\pi} \sim N(\mu, 1/\tau) \sim NormalGamma(\mu_0, \lambda, \alpha, \beta)$
  - $T \sim Categorical(\boldsymbol{p}) \sim Dirichlet(\boldsymbol{\rho})$
- Action selection: Thompson sampling
- Monte-Carlo tree search

Main Algorithm

OnlinePlanning (s : *state*, T : *tree*, H : *max horizon*)
Initialize $(\mu_{s,0}, \lambda_s, \alpha_s, \beta_s)$ for each $s \in S$
Initialize $\boldsymbol{\rho}_{s,a}$ for each $s \in S$ and $a \in A$
repeat
    DNG-MCTS (s, T, H)
until *resource budgets reached*
return ThompsonSampling $(s, H, False)$

DNG-MCTS (s : *state*, T : *tree*, h : *horizon*)
if $h = 0$ **or** s *is terminal* **then**
    return 0
else if *node* (s, h) *is not in tree* T **then**
    Add node (s, h) to T
    Play rollout policy by simulation for h steps
    Observe the outcome r
    return r
else
    $a \leftarrow$ **ThompsonSampling** $(s, h, True)$
    Execute a by simulation
    Observe next state s' and reward $R(s, a)$
    $r \leftarrow R(s,a) + \gamma$ **DNG-MCTS** $(s', T, h-1)$
    $\alpha_s \leftarrow \alpha_s + 0.5$
    $\beta_s \leftarrow \beta_s + (\lambda_s(r - \mu_{s,0})^2/(\lambda_s + 1))/2$
    $\mu_{s,0} \leftarrow (\lambda_s \mu_{s,0} + r)/(\lambda_s + 1)$
    $\lambda_s \leftarrow \lambda_s + 1$
    $\rho_{s,a,s'} \leftarrow \rho_{s,a,s'} + 1$
    return r

ThompsonSampling (s : *state*, h : *horizon*, *sampling* : *boolean*)
foreach $a \in A$ **do**
    $q_a \leftarrow$ **QValue** $(s, a, h, sampling)$
return $\operatorname{argmax}_a q_a$

QValue (s : *state*, a : *action*, h : *horizon*, *sampling* : *boolean*)
$r \leftarrow 0$
foreach $s' \in S$ **do**
    if $sampling = True$ **then**
        Sample $w_{s'}$ according to $Dir(\boldsymbol{\rho}_{s,a})$
    else
        $w_{s'} \leftarrow \rho_{s,a,s'} / \sum_{n \in S} \rho_{s,a,n}$
    $r \leftarrow r + w_{s'}$ **Value** $(s', h-1, sampling)$
return $R(s,a) + \gamma r$

Value (s : *state*, h : *horizon*, *sampling* : *boolean*)
if $h = 0$ **or** s *is terminal* **then**
    return 0
else
    if $sampling = True$ **then**
        Sample (μ, τ) according to $NormalGamma(\mu_{s,0}, \lambda_s, \alpha_s, \beta_s)$
        return μ
    else
        return $\mu_{s,0}$

Experiments

- MDP benchmark problems (cost based)
  - Canadian Traveler Problem
  - Race Track Problem
  - Sailing Problem
- Experiments
  - Run from current state for a quantity of iterations
  - Apply the best action according to the returned values
  - Repeat the loop until terminating conditions
  - Report the total discounted cost/reward

Canadian Traveler Problem

- CTP
  - A path finding problem
  - Imperfect information over a graph
  - Edges may be blocked with given prior probabilities
  - Modeled as a deterministic POMDP
  - Transformed to an MDP
  - Belief size: $n \times 3^m$

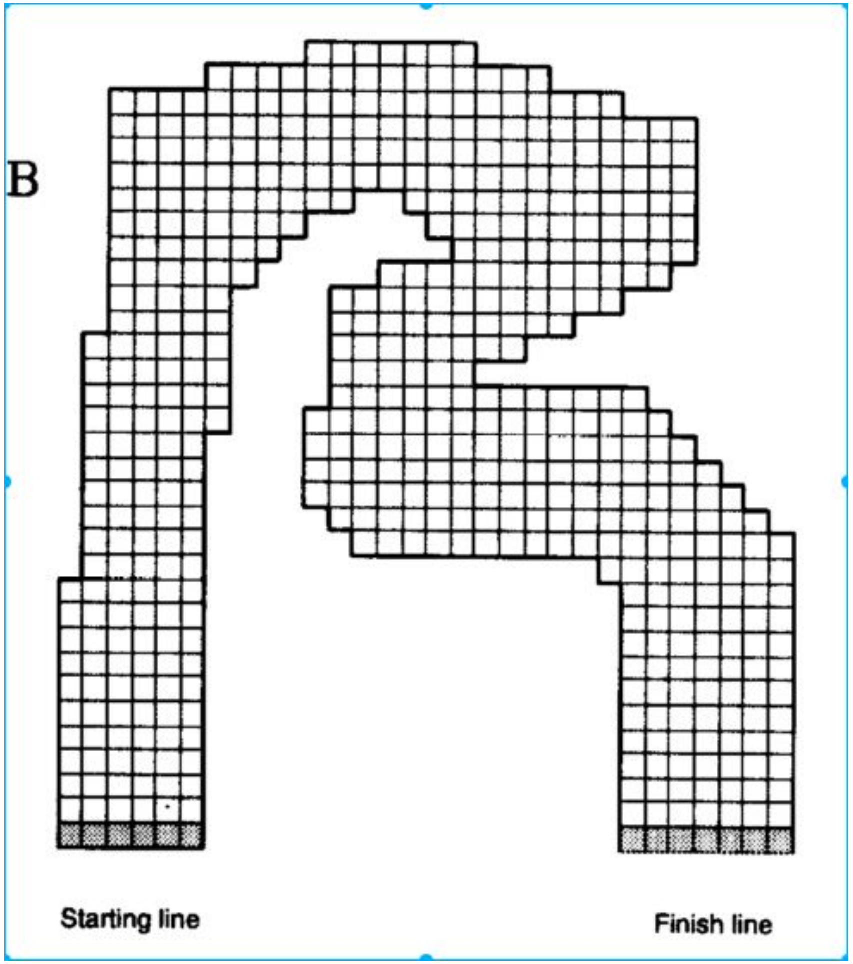
Canadian Traveler Problem (cont.)

Table 1: CTP problems with 20 nodes. The second column indicates the belief size of the transformed MDP for each problem instance. UCTB and UCTO are the two domain-specific UCT implementations [18]. DNG-MCTS and UCT run for 10,000 iterations. Two groups of rollout policy are tested: random policy and optimistic policy. Boldface fonts are best in whole table; gray cells show best among domain-independent implementations for each group. The data of UCTB, UCTO and UCT are taken form [16].

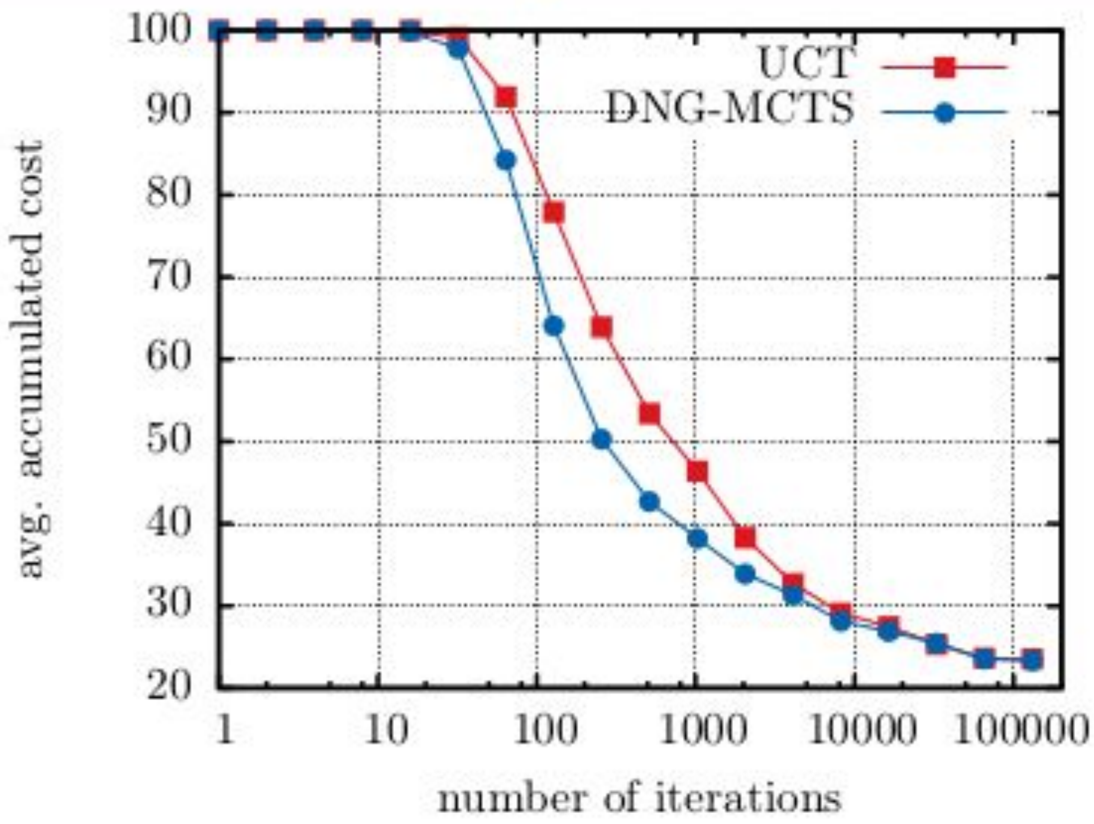
		domain-specific UCT		random rollout policy		optimistic rollout policy	
prob.	belief	UCTB	UCTO	UCT	DNG	UCT	DNG
20-1	20×3^{49}	210.7±7	**169.0±6**	216.4±3	223.9±4	180.7±3	177.1±3
20-2	20×3^{49}	176.4±4	**148.9±3**	178.5±2	178.1±2	160.8±2	155.2±2
20-3	20×3^{51}	150.7±7	**132.5±6**	169.7±4	159.5±4	144.3±3	140.1±3
20-4	20×3^{49}	264.8±9	**235.2±7**	264.1±4	266.8±4	238.3±3	242.7±4
20-5	20×3^{52}	123.2±7	**111.3±5**	139.8±4	133.4±4	123.9±3	122.1±3
20-6	20×3^{49}	165.4±6	**133.1±3**	178.0±3	169.8±3	167.8±2	141.9±2
20-7	20×3^{50}	191.6±6	**148.2±4**	211.8±3	214.9±4	174.1±2	166.1±3
20-8	20×3^{51}	160.1±7	**134.5±5**	218.5±4	202.3±4	152.3±3	151.4±3
20-9	20×3^{50}	235.2±6	**173.9±4**	251.9±3	246.0±3	185.2±2	180.4±2
20-10	20×3^{49}	180.8±7	**167.0±5**	185.7±3	188.9±4	178.5±3	170.5±3
total		1858.9	**1553.6**	2014.4	1983.68	1705.9	1647.4

Race Track Problem

- RaceTrack
  - A car starts in a set of initial states
  - Moves towards the goal
  - Choose to accelerate to one of the eight directions
  - Probability of 0.9 to succeed
  - Probability 0.1 to fail on its acceleration
  - State space size: 22534



Race Track Problem (cont.)

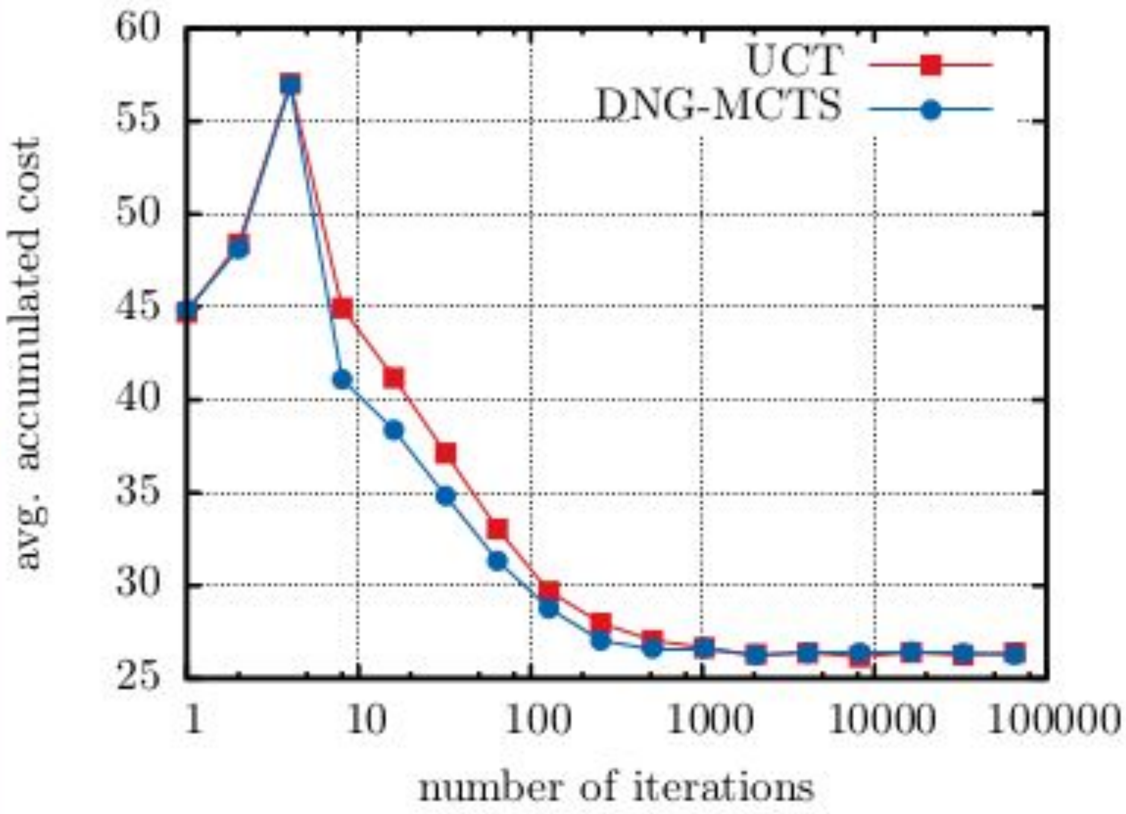


(a) Racetrack-barto-big with random policy

Sailing Problem

- Sailing
  - A sailboat navigates to a destination
  - Direction of the wind changes over time
  - Choose at each grid location a neighbour location to move to
  - The goal is to reach the destination as quickly as possible
  - State space size: 80000

Sailing Problem (cont.)



(b) Sailing-100 × 100 with random policy

Conclusion & Future Work

- DNG-MCTS
  - Bayesian Approach
  - Thompson Sampling
  - Monte-Carlo Tree Search
  - Competitive results comparing to UCT
- Future Work
  - Test in POMDP and Bayes RL domain
  - Motion planning problem in robotics
    - Physical simulator
    - Monte-Carlo method
    - Bayesian approach