# MCTS Based on Simple Regret

David Tolpin, Solomon Eyal Shimony

Ben-Gurion University of the Negev
Beer Sheva, Israel

February 22, 2012

# Hard to solve search problems

Search problems are often hard too solve in practice when:

- search space is extremely large;
- *and* good heuristics are unknown.

**Easier** to solve:

- Chess — search space size is manageable ($10^{50}$).
- Timetabling — good heuristics.

**Hard** to solve:

- Compute Go ($10^{180}$), Poker ($10^{70}$).
- Canadian Traveller Problem.

# MCTS

**M**onte **C**arlo **T**ree **S**earch helps in large search spaces.

- Starts with the root only.

**Adaptive** MCTS samples ‘good’ moves more frequently, but sometimes **explores** new directions.

# MCTS

**M**onte **C**arlo **T**ree **S**earch helps in large search spaces.

- Starts with the root only.
- Repeats:
  1. **Selection:** select a branch to explore.

**Adaptive** MCTS samples 'good' moves more frequently, but sometimes **explores** new directions.

# MCTS

**M**onte **C**arlo **T**ree **S**earch helps in large search spaces.

- Starts with the root only.
- Repeats:
  1. **Selection:** select a branch to explore.
  2. **Expansion:** adds children of the leaf to the stored tree.

**Adaptive** MCTS samples 'good' moves more frequently,
but sometimes **explores** new directions.

# MCTS

**M**onte **C**arlo **T**ree **S**earch helps in large search spaces.

- Starts with the root only.
- Repeats:
    1. **Selection:** select a branch to explore.
    2. **Expansion:** adds children of the leaf to the stored tree.
    3. **Simulation:** continues search (using a simple strategy) until a goal state is reached.

**Adaptive** MCTS samples ‘good’ moves more frequently,
but sometimes **explores** new directions.

# MCTS

**M**onte **C**arlo **T**ree **S**earch helps in large search spaces.

- Starts with the root only.
- Repeats:
  1. **Selection:** select a branch to explore.
  2. **Expansion:** adds children of the leaf to the stored tree.
  3. **Simulation:** continues search (using a simple strategy) until a goal state is reached.
  4. **Backpropagation:** values of each stored node are updated.

**Adaptive** MCTS samples 'good' moves more frequently, but sometimes **explores** new directions.

# Multi-armed Bandit Problem and UCB

Multi-armed Bandit Problem:

- We are given a set of $K$ arms.
- Each arm can be pulled multiple times.
- The reward is drawn from an **unknown** (but normally *stationary* and *bounded*) distribution.
- The **total reward** must be maximized.

**UCB** is near-optimal for MAB — solves *exploration/exploitation* tradeoff.

- pulls an arm that maximizes **U**pper **C**onfidence **B**ound: $b_i = \overline{X}_i + \sqrt{\frac{c \log(n)}{n_i}}$
- the cumulative regret is $O(\log n)$.

# UCT

UCT (**U**pper **C**onfidence Bounds applied to **T**rees) is based on UCB.

- Adaptive MCTS.
- Applies the UCB selection scheme at each step of the rollout.
- Demonstrated good performance in Computer Go (MoGo, CrazyStone, Fuego, Pachi, ...) as well as in other domains.

However, the first step of a rollout is different:

- The purpose of MCTS is to choose an action with the greatest utility.
- Therefore, the **simple regret** must be minimized.

# SRCR

**S**imple **R**egret followed by **C**umulative **R**egret.

- Minimizes **simple regret** at the **first step**.
- Continues with UCT from the **second step on**.

```
procedure Rollout(node, depth=1)
    if IsLeaf(node, depth) then
        return 0
    else
        if depth=1 then action ← FirstAction(node)
        else action ← NextAction(node)
        next-node ← NextState(node, action)
        reward ← Reward(node, action, next-node)
                    + Rollout(next-node, depth+1)
        UpdateStats(node, action, reward)
```

# Sampling for Simple Regret

Sampling schemes for miniminizing the simple regret:

1. $\varepsilon$-greedy sampling.
2. a modified version of UCB (worse for cumulative, better for simple regret).
3. VOI-based sampling.

# Sampling for Simple Regret

Sampling schemes for miniminizing the simple regret:

1. $\varepsilon$-greedy sampling.
2. a modified version of UCB (worse for cumulative, better for simple regret).
3. VOI-based sampling.

- 1, 2 — heuristic selection criterion, theoretical upper bounds can be obtained.

# Sampling for Simple Regret

Sampling schemes for miniminizing the simple regret:

1. $\varepsilon$-greedy sampling.
2. a modified version of UCB (worse for cumulative, better for simple regret).
3. VOI-based sampling.

- 1, 2 — heuristic selection criterion, theoretical upper bounds can be obtained.
- 3 — based on principles of *Rational Metareasoning*, but harder to analyze.

# Heuristic sampling schemes

$\varepsilon$-greedy:

- Pulls the empirically best arm with probability $\varepsilon$.
- Any other arm with probability $\frac{1-\varepsilon}{K-1}$.
- Exhibits exponentially decreasing simple regret.
- Uniform sampling when $\varepsilon = \frac{1}{K}$, much better when $\varepsilon = \frac{1}{2}$.

# Heuristic sampling schemes

$\varepsilon$-greedy:

- Pulls the empirically best arm with probability $\varepsilon$.
- Any other arm with probability $\frac{1-\varepsilon}{K-1}$.
- Exhibits exponentially decreasing simple regret.
- Uniform sampling when $\varepsilon = \frac{1}{K}$, much better when $\varepsilon = \frac{1}{2}$.

$UCB_{\sqrt{\cdot}}$ ($\sqrt{\cdot}$ instead of log):

- Pulls arm $i$ that maximizes $b_i = \overline{X}_i + \sqrt{\frac{c\sqrt{n}}{n_i}}$.
- Exhibits superpolynomially decreasing simple regret.

# VOI-aware sampling

- Chooses an action with the maximum VOI estimate.
- Estimates the VOI based on bounds on:
  - the probability that one or more rollouts will make another action appear better than the current best;
  - the gain that may be incurred if such a change occurs.

$$
\begin{aligned}
VOI_\alpha &\approx \frac{\overline{X}_\beta}{n_\alpha + 1} \exp\left(-2(\overline{X}_\alpha - \overline{X}_\beta)^2 n_\alpha\right) \\
VOI_i &\approx \frac{1 - \overline{X}_\alpha}{n_i + 1} \exp\left(-2(\overline{X}_\alpha - \overline{X}_i)^2 n_i\right), \quad i \neq \alpha \\
\text{where} \quad & \alpha = \arg\max_i \overline{X}_i, \quad \beta = \arg\max_{i,\, i \neq \alpha} \overline{X}_i
\end{aligned}
$$

# Simple regret in MAB

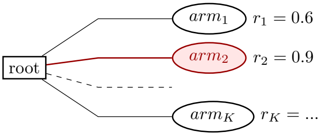



Fig. 1: Search tree

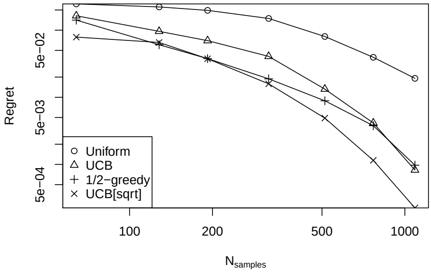



Fig. 2: Regret vs # of samples (32 arms)

- For smaller numbers of samples, $\frac{1}{2}$-greedy achieves the best performance.
- For larger numbers of samples, UCB$_{\sqrt{\cdot}}$ outperforms $\frac{1}{2}$-greedy.
- A combination of $\frac{1}{2}$-greedy and UCB$_{\sqrt{\cdot}}$ dominates UCB over the whole range.

# Monte Carlo tree search

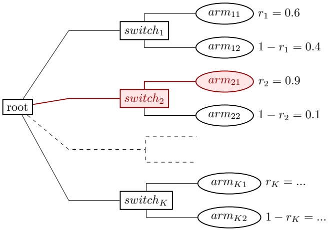



Fig. 1: Search tree

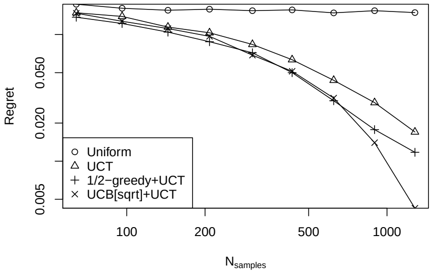



Fig. 2: Regret vs # of samples (16 switches)

- Either $\frac{1}{2}$-greedy+UCT or UCB$_{\sqrt{\cdot}}$+UCT gives the lowest regret.
- UCB$_{\sqrt{\cdot}}$+UCT dominates UCT everywhere except for small numbers of instances.
- The advantage of both $\frac{1}{2}$-greedy+UCT and UCB$_{\sqrt{\cdot}}$+UCT grows with the number of arms.

# Sailing domain

- A square lake.
- A sailboat has to find the shortest path between corners.
- The wind changes randomly.

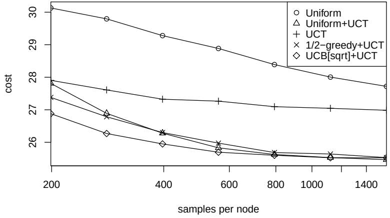



Fig. 1: Median cost

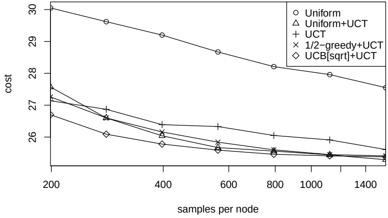
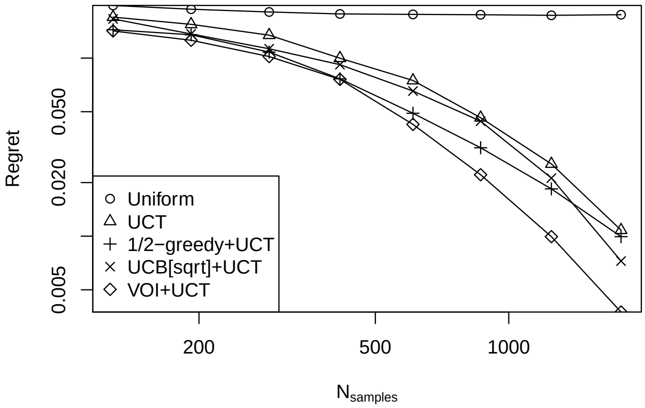



Fig. 2: Minimum cost

- UCT is always worse than $\frac{1}{2}$-greedy+UCT or UCB$_{\sqrt{\cdot}}$+UCT.
- UCT is sensitive to the value of $c$: the median cost is much higher than the minimum cost.

# VOI-aware MCTS




- The experiments were performed on randomly generated trees.
- VOI+UCT, the scheme based on a VOI estimate, outperforms all other sampling schemes in this example.

# Summary

- Improved MCTS scheme, SRCR, introduced.

# Summary

- Improved MCTS scheme, SRCR, introduced.
- SRCR performs better than unmodified UCT.

# Summary

- Improved MCTS scheme, SRCR, introduced.
- SRCR performs better than unmodified UCT.
- VOI-aware sampling for minimizing sampling regret proposed.

# Summary

- Improved MCTS scheme, SRCR, introduced.
- SRCR performs better than unmodified UCT.
- VOI-aware sampling for minimizing sampling regret proposed.
- Better sampling schemes can be developed based on principles of Rational Metareasoning.

## Summary

- Improved MCTS scheme, SRCR, introduced.
- SRCR performs better than unmodified UCT.
- VOI-aware sampling for minimizing sampling regret proposed.
- Better sampling schemes can be developed based on principles of Rational Metareasoning.
- UCT is not well understood. Better understanding will help in developing efficient MCTS schemes and adapting to different domains.

Thank you!