MCTS BASED ON SIMPLE REGRET AAAI 2012

Multi-Armed Bandits

- A set of K arms.
- Each arm can be pulled multiple times.
- times.

 When the ith arm is pulled, a random reward X_i is encountered.
- Simple regret: the reward of the last pull only is collected.
- Cumulative regret: all rewards are accumulated.

UCB AND **UCT**

• **UCB**(c) pulls arm i that maximizes upper confidence bound b_i on the reward: $c \log(n)$

$$b_i = \overline{X}_i + \sqrt{rac{c \log(n)}{n_i}}$$

- UCB is nearly optimal in minimizing the *cumulative regret*.
- **UCT** extends UCB to MCTS by invoking UCB in every node of a rollout.

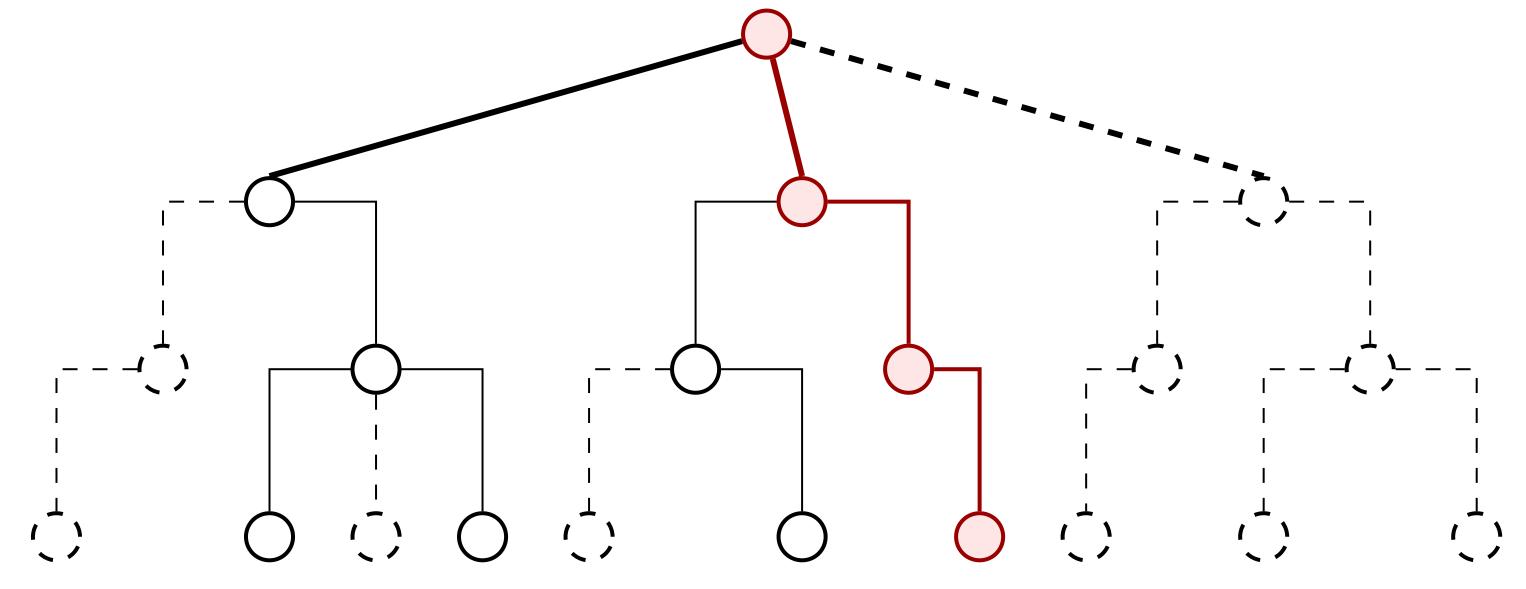
METAREASONING

- A problem-solving agent can perform base-level actions from a known set $\{A_i\}$.
- Before committing to an action, the agent may perform a sequence of meta-level deliberation actions from a set $\{S_i\}$.
- At any given time there is a base-level action A_{α} that maximizes the agent's *expected utility*.
- The value of information VOI_j is the expected difference between the expected utilities of the new and the old selected base-level action after meta-level action S_j is taken.
- The agent selects a meta-level action that maximizes the VOI, or A_{α} if no meta-level action has positive VOI.

ACKNOWLEDGMENTS

- IMG4 Consortium under the MAGNET program of the Israeli Ministry of Trade and Industry
- Israel Science Foundation grant 305/09
- Lynne and William Frankel Center for Computer Sciences
- Paul Ivanier Center for Robotics Research and Production Management

Monte-Carlo Sampling in Trees



- MCTS performs multiple *rollouts* to partially explore the search space.
- At the current root node, the sampling is aimed at finding the first move to perform: minimizing the simple regret is more appropriate at the root node.
- Deeper in the tree, minimizing cumulative regret results in a more precise estimate of the value of the state.
- An improvement over UCT can be achieved by combining different sampling schemes on the first step and during the rest of a rollout.

MAIN RESULTS

The SR+CR MCTS Scheme

- Selects an action at **the current root** suitable for minimizing the simple regret.
- Then selects actions according to UCB, that approximately minimizes the cumulative regret.

```
1 Rollout(node, depth=1)
2 if IsLeaf(node, depth)
3 return 0
4 else
5 if depth=1 then action 	— FirstAction(node)
6 else action 	— NextAction(node)
7 next 	— NextState(node, action)
8 reward 	— Reward(node, action, next)
9 + Rollout(next, depth+1)
10 UpdateStats(node, action, reward)
11 return reward
```

Sampling for Simple Regret

- 1. ε -greedy sampling $\left(\varepsilon = \frac{1}{2}\right)$.
- 2. Modified version of **UCB** (optimized for *simple regret*).
- 3. **VOI-aware** sampling:

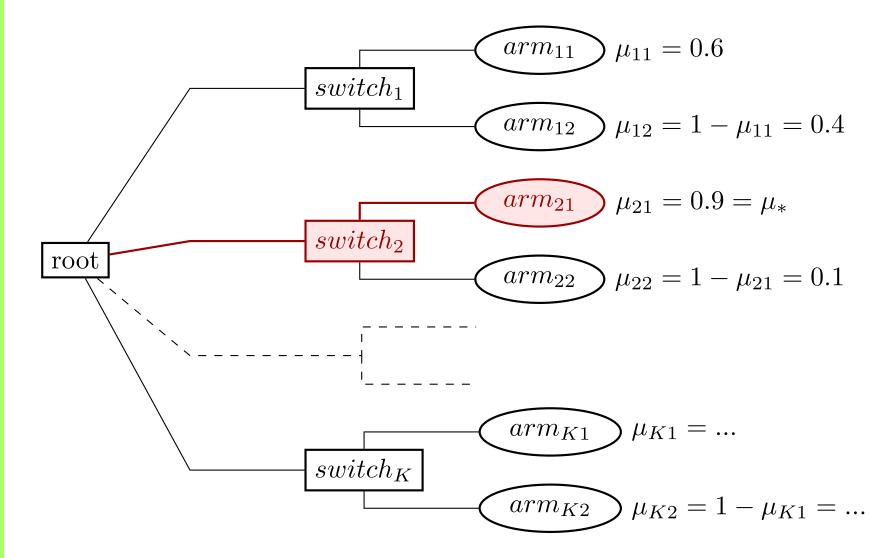
$$VOI_lpha pprox rac{\overline{X}_eta}{n_lpha + 1} \exp\Bigl(-2(\overline{X}_lpha - \overline{X}_eta)^2 n_lpha\Bigr)$$

$$VOI_i pprox rac{1-\overline{X}_lpha}{n_i+1} \exp\Bigl(-2(\overline{X}_lpha-\overline{X}_i)^2 n_i\Bigr), \ i
eq lpha$$

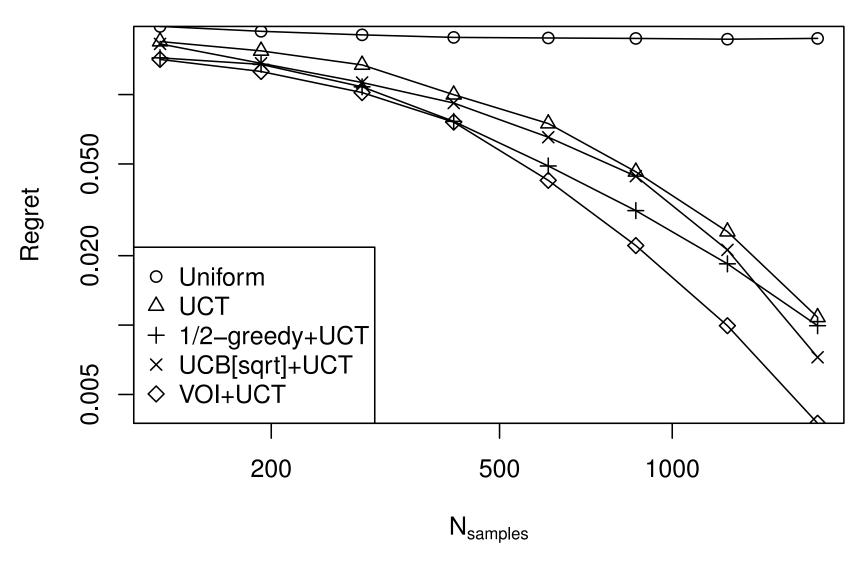
EXPERIMENTS

- SR+CR outperforms UCT.
- SR+UCT(c) is less dependent on tuning of the exploration factor c.

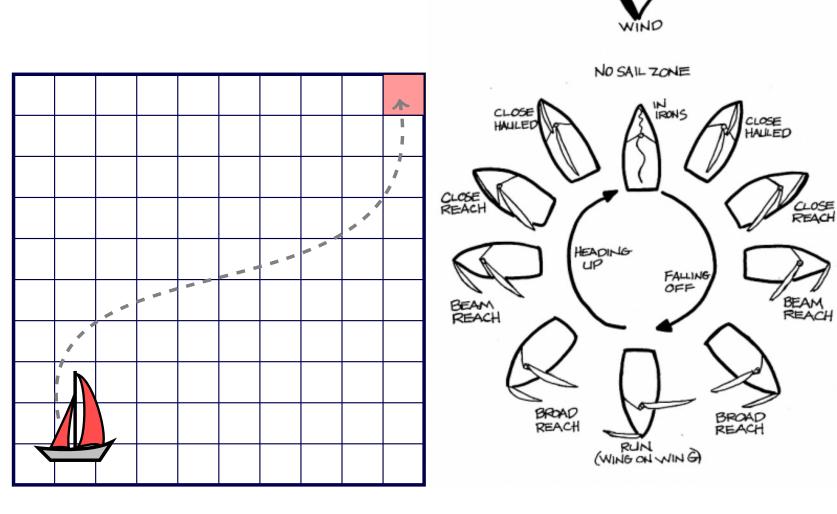
Random Trees



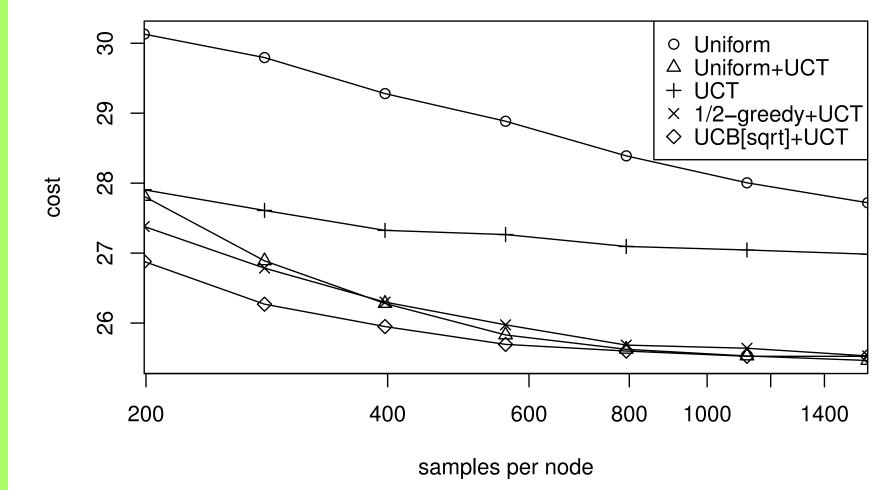
Regret vs. number of samples:



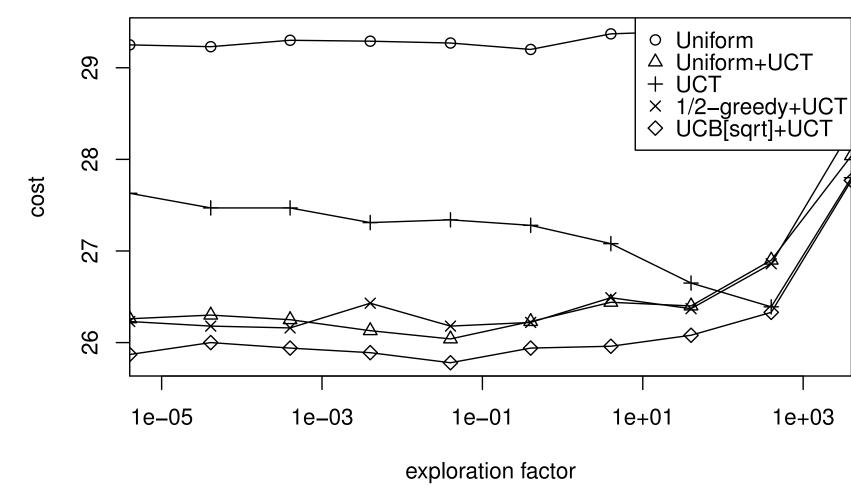
Sailing Domain



Path cost vs. number of samples:



Path cost vs. exploration factor:



CONTRIBUTIONS

- Improved MCTS scheme SR+CR.
- SR+CR performs better than unmodified UCT.
- VOI-aware sampling for minimizing simple regret.

FUTURE WORK

- Rational metareasoning in MCTS: theory and VOI estimates.
- Better sampling for non-root nodes.
- Application to Computer Go and other complex domains.

