

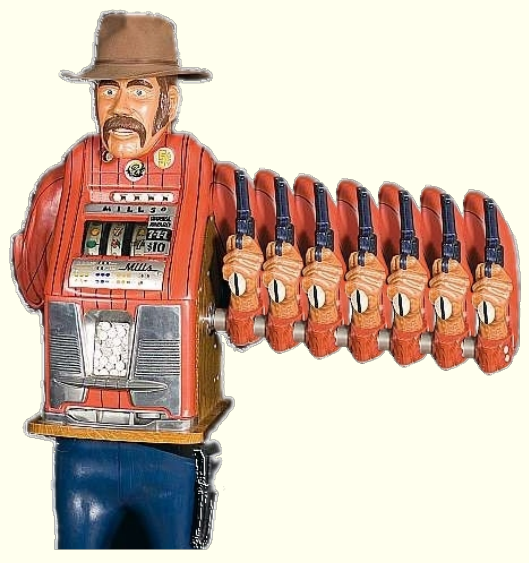
# MCTS BASED ON SIMPLE REGRET

## AAAI 2012

David Tolpin, Solomon Eyal Shimony  
Ben Gurion University of the Negev,  
Beer Sheva, Israel

### MULTI-ARMED BANDITS

- A set of  $K$  arms.
- Each arm can be pulled multiple times.
- When the  $i$ th 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:  

$$b_i = \bar{X}_i + \sqrt{\frac{c \log(n)}{n_i}}$$
- UCB is nearly optimal in minimizing the *cumulative regret*.
- UCT** extends UCB to MCTS by invoking UCB in every node of a rollout.

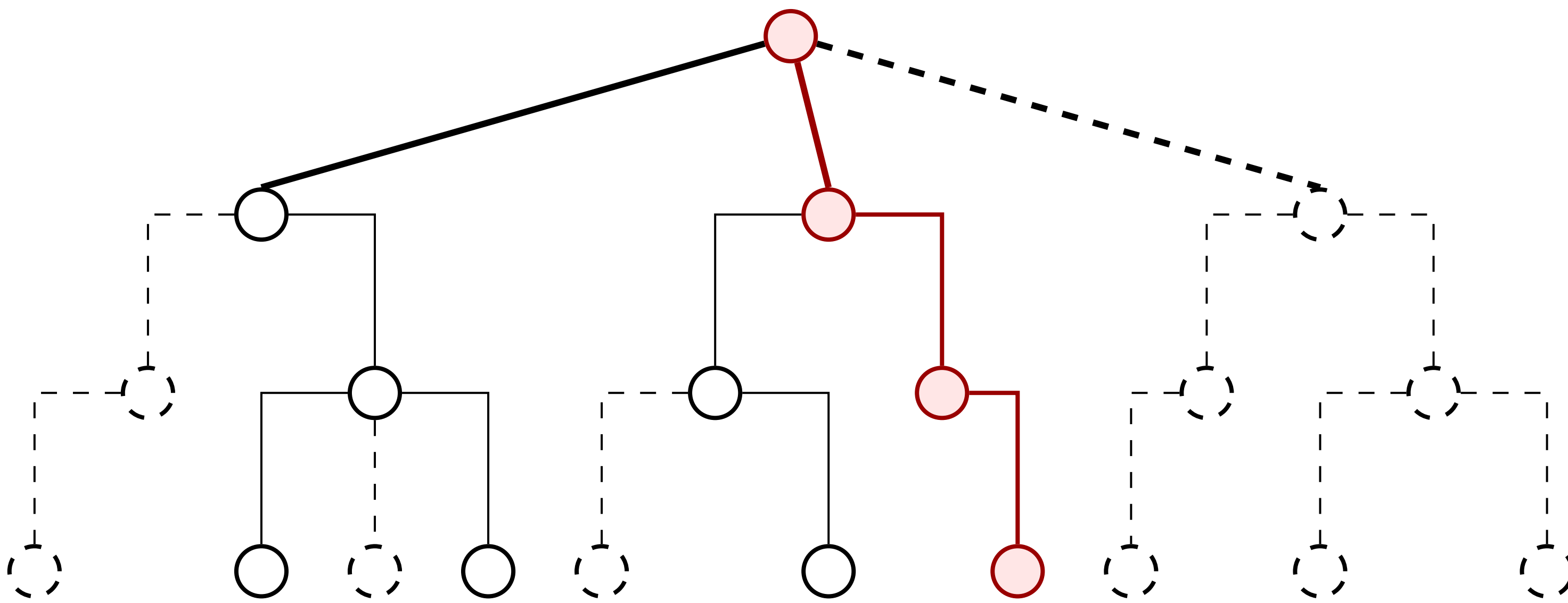
### METAREASONING

- 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  $\{S_j\}$ .
- At any given time there is a base-level action  $A_\alpha$  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  $A_\alpha$  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

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

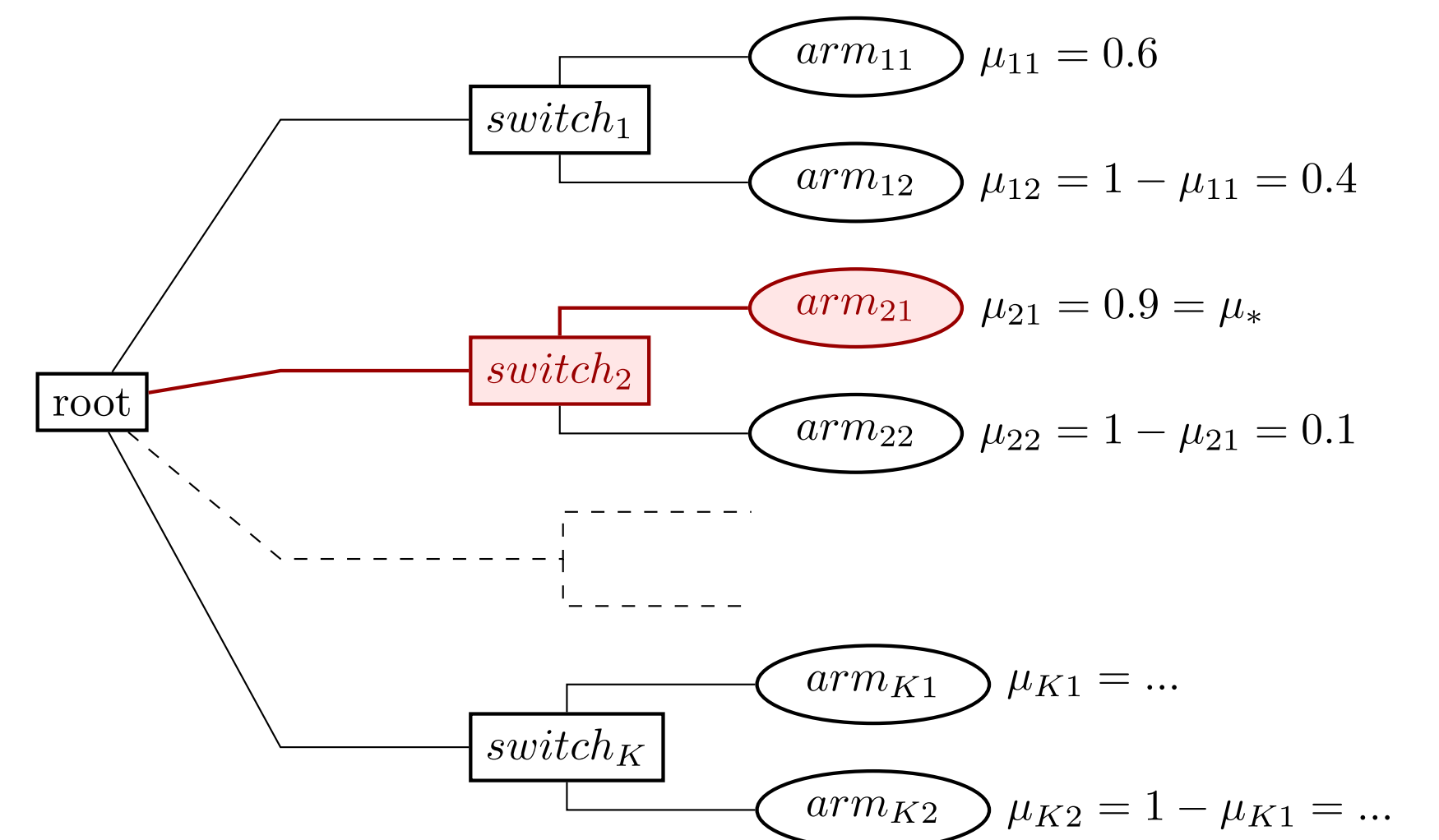
$$VOI_\alpha \approx \frac{\bar{X}_\beta}{n_\alpha + 1} \exp\left(-2(\bar{X}_\alpha - \bar{X}_\beta)^2 n_\alpha\right)$$

$$VOI_i \approx \frac{1 - \bar{X}_\alpha}{n_i + 1} \exp\left(-2(\bar{X}_\alpha - \bar{X}_i)^2 n_i\right), i \neq \alpha$$

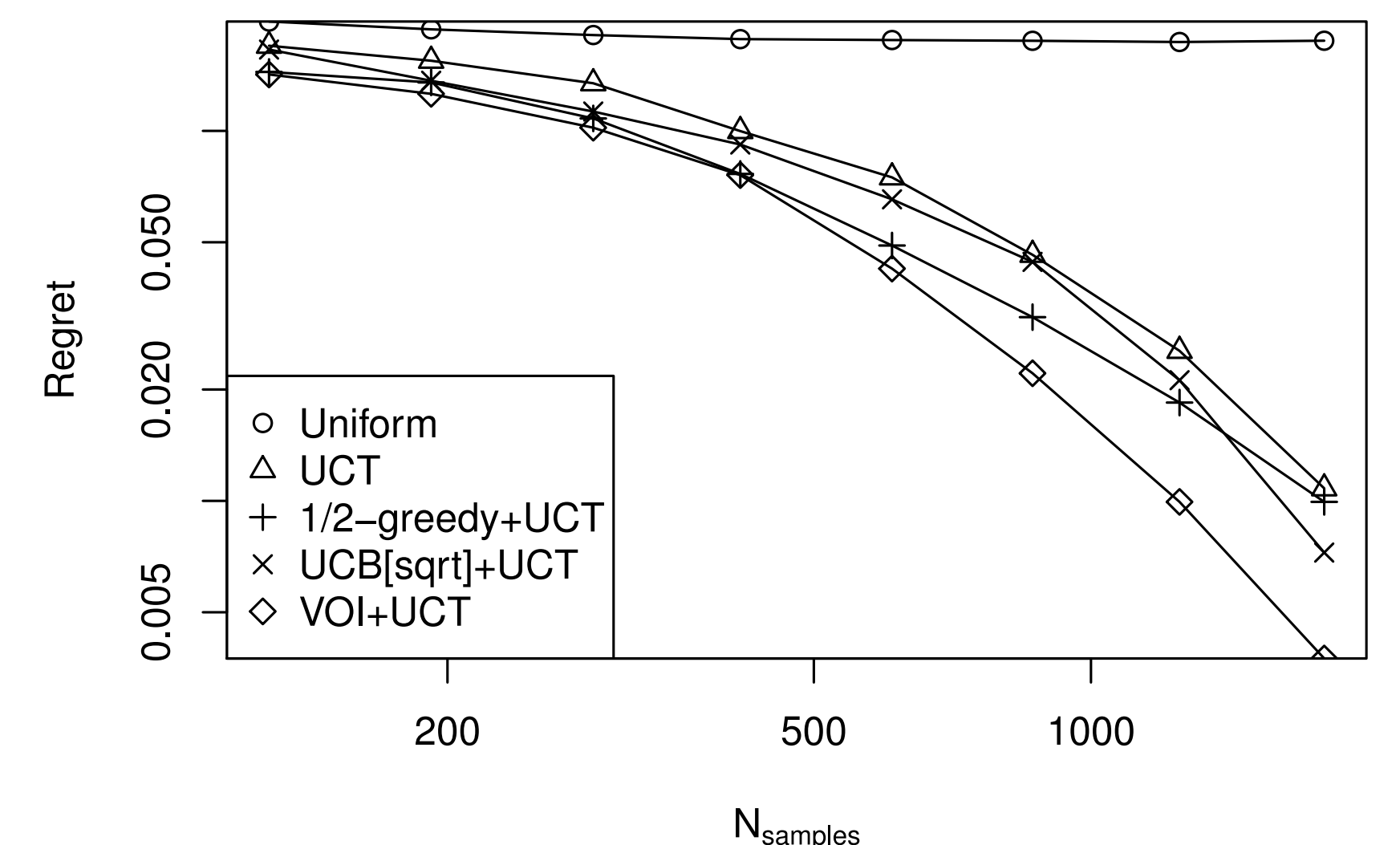
### EXPERIMENTS

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

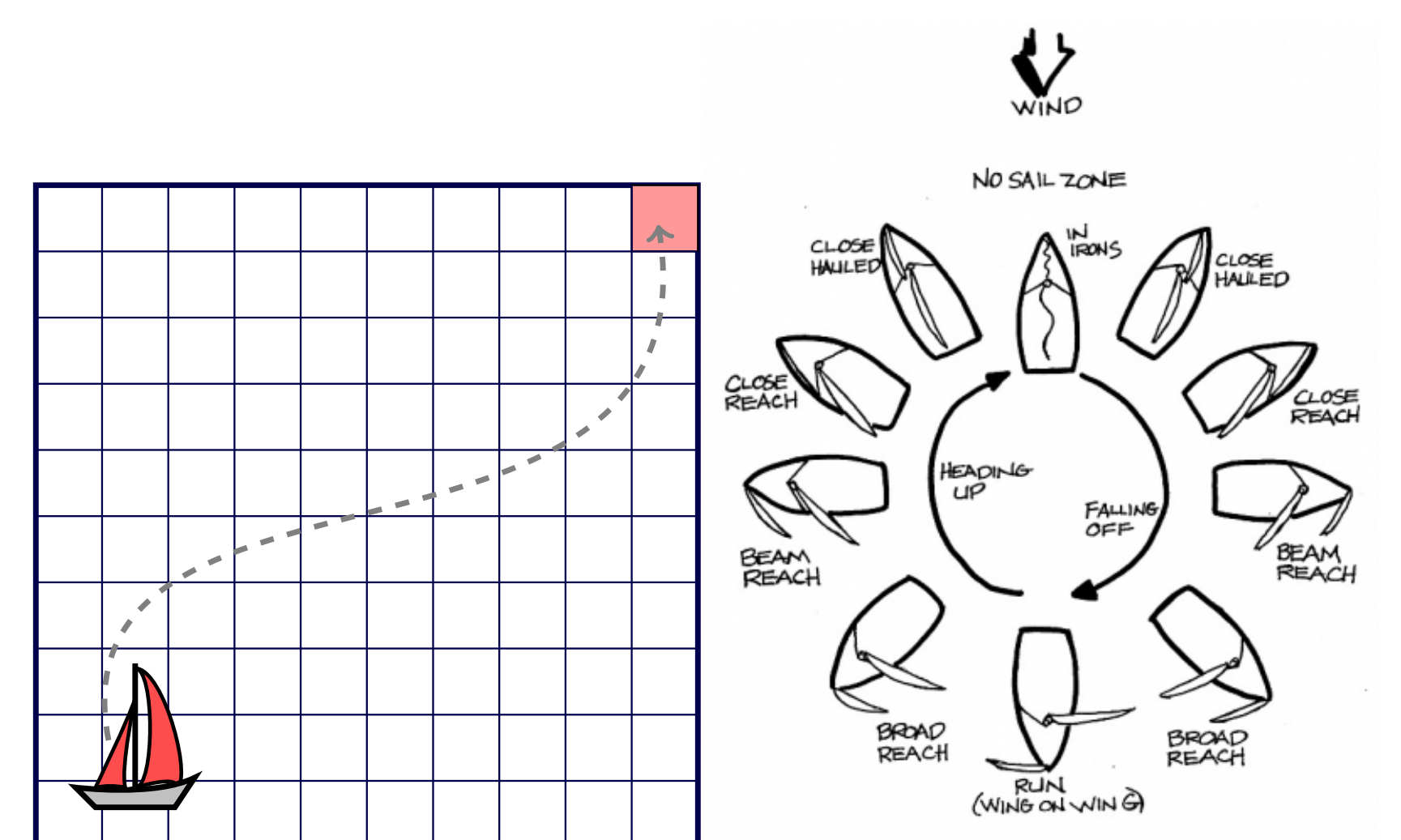
#### Random Trees



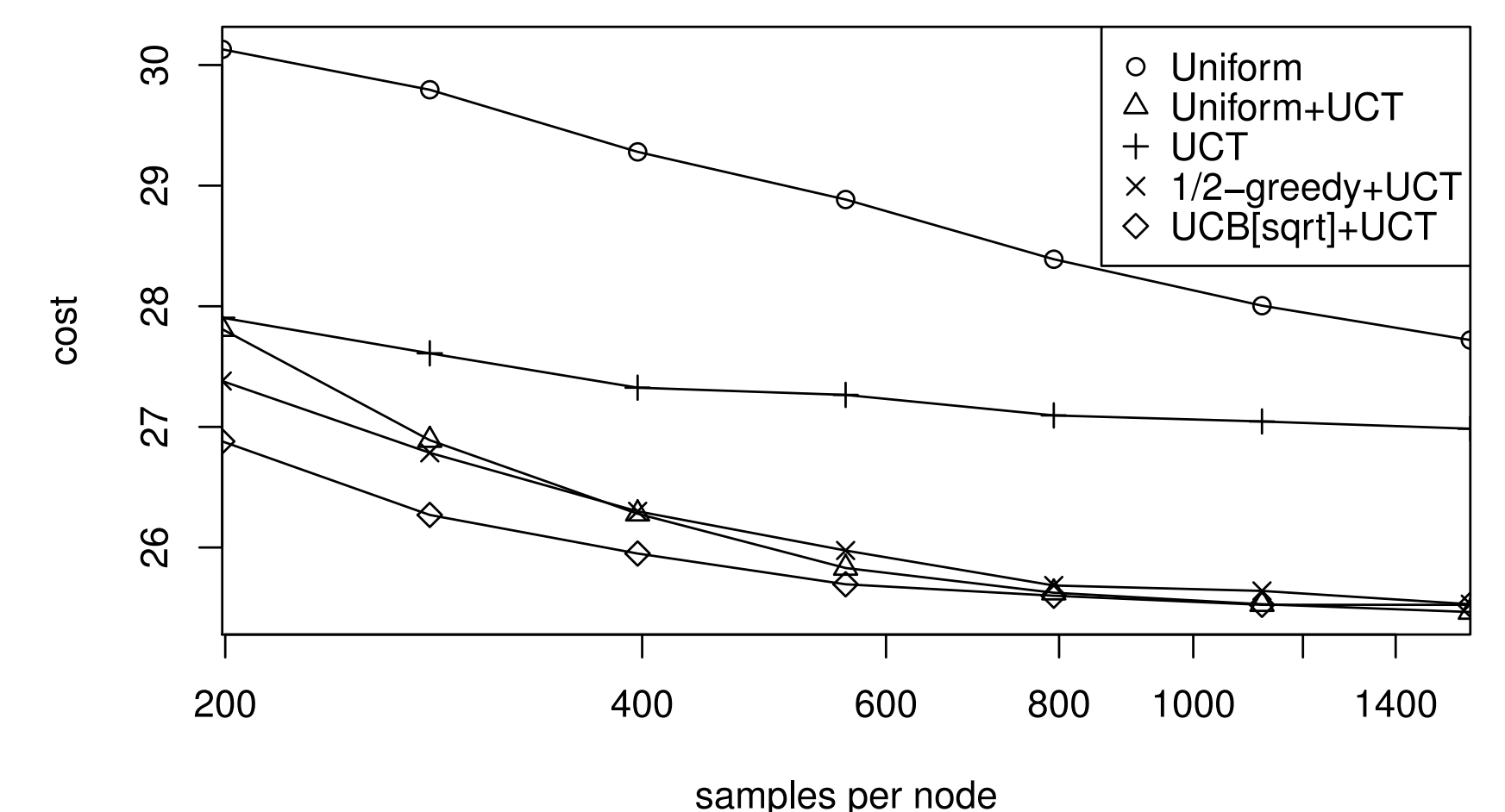
#### Regret vs. number of samples:



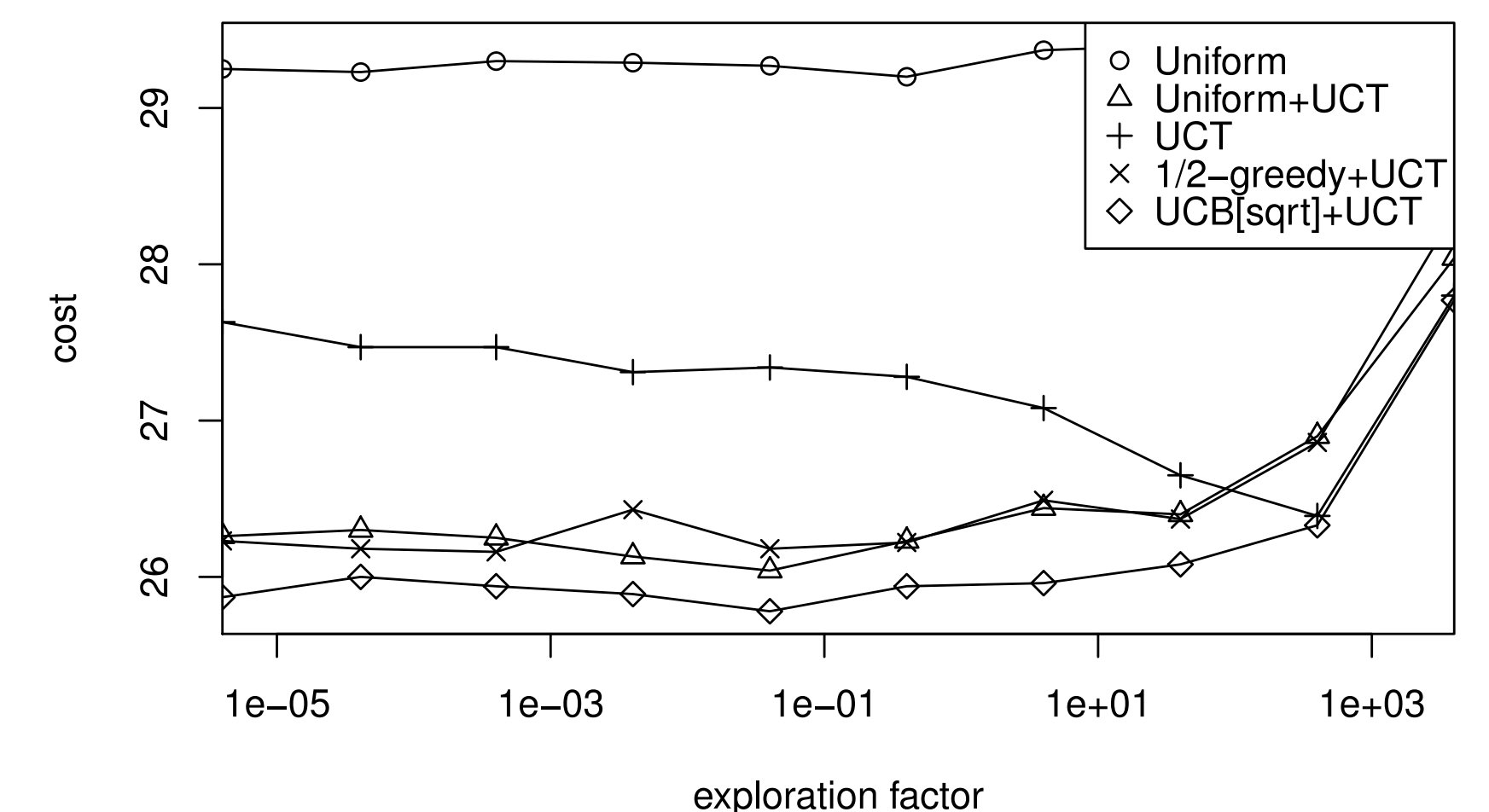
#### Sailing Domain



#### 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.

