Doing Better Than UCT: Rational Monte Carlo Sampling in Trees

David Tolpin, Solomon Eyal Shimony
Department of Computer Science,
Ben-Gurion University of the Negev, Beer Sheva, Israel
{tolpin,shimony}@cs.bgu.ac.il

August 18, 2011

Abstract

UCT, a state-of-the art algorithm for Monte Carlo tree sampling (MCTS), is based on UCB, a sampling policy for the Multi-armed Bandit Problem (MAB) that minimizes the accumulated regret. However, MCTS differs from MAB in that only the final choice, rather than all arm pulls, brings a reward, that is, the simple regret, as opposite to the cumulative regret, must be minimized. This ongoing work aims at applying meta-reasoning techniques to MCTS, which is non-trivial. We begin by introducing policies for multi-armed bandits with lower simple regret than UCB, and an algorithm for MCTS which combines cumulative and simple regret minimization and outperforms UCT. We also develop a sampling scheme loosely based on a myopic version of perfect value of information. Finite-time and asymptotic analysis of the policies is provided, and the algorithms are compared empirically.

1 Introduction and definitions

Monte-Carlo tree sampling, and especially a version based on the UCT formula [5] appears in numerous search applications, such as [3]. Although these methods are shown to be successful empirically, most authors appear to be using the UCT formula "because it has been shown to be successful in the past", and "because it does a good job of trading off exploration and exploitation". While the latter statement may be correct for the multi-armed bandit and for the UCB method [1], we argue that it is inappropriate for search. The problem is not that UCT does not work; rather, we argue that a simple reconsideration from basic principles can result in schemes that outperform UCT. This is demonstrated both theoretically and empirically.

In the Multi-armed Bandit problem we have a set of K arms. Each arm can be pulled multiple times. When the ith arm is pulled, a random reward X_i from an unknown stationary distribution is returned. The reward is bounded between 0 and 1.

The simple regret of a sampling policy for the Multi-armed Bandit Problem is the expected difference between the best expected reward μ_* and the expected reward μ_j of the arm with the best sample mean $\overline{X}_j = \max_i \overline{X}_i$:

$$\mathbb{E}[R] = \sum_{j=1}^{K} \Delta_j \Pr(\overline{X}_j = \max_i \overline{X}_i) \tag{1}$$

where $\Delta_j = \mu_* - \mu_j$. Strategies that minimize the simple reward are called pure exploration strategies [2].

Such strategies are used to select the best arm, and, by extension, the best action in Monte Carlo tree search. Monte Carlo tree search is used to solve Markov Decision Processes (MDP) approximately.

An MDP is defined by the set of states S, the set of actions A, the transition table $T(s, a, s')$, the reward table $R(s, a, s')$, the initial state s and the goal state t: (S, A, T, R, s, t) [6]. Monte Carlo tree search explores an MDP by *rollouts*—trajectories from the current state to a state in which a termination condition is satisfied (either the goal state, or a cutoff state for which the reward is evaluated approximately). The UCB algorithm (that attempts to minimize the cumulative regret) [1] had been extended into the tree search sampling scheme known as UCT [5].

UCT-driven search achieves good results in many search problems. At each search step, the algorithm allocates Monte Carlo rollouts to choose the best action, thereby (approximately) minimizing the cumulative regret. The core issue is that in search (especially for adversarial search and for "games against nature" - optimizing behavior under uncertainty) the goal is typically to either find a good (or optimal) strategy, or even to find the best first action of such a policy. Once such an action is discovered, it is not beneficial to further sample that action, "exploitation" is thus meaningless for search problems. Finding a good first action is closer to the pure exploration variant, as seen in the selection problem [2, 7]. In the selection problem, it is much better to minimize the *simple* regret. However, the simple and the cumulative regret cannot be minimized simultaneously; moreover, [2] shows that in many cases the smaller the cumulative regret, the greater the simple regret.

UCT performance can thus be improved by combining UCB with a sampling scheme that minimizes the simple regret of selecting an action at the current root node. Indeed, the algorithm must select an action with the minimum regret *given the assumption that after performing the selected action the algorithm performs optimally*, which corresponds to maximizing the value of partial information [6]. Therefore, an improved allocation scheme would

- maximize the value of partial information by sampling actions to minimize the **simple** regret of the selection at the current root node, and
- as the goal of sampling in deeper tree nodes is estimating the value of a node, rather than selection, it makes sense to minimize the **cumulative** regret of the rollouts from the second step onwards.

The ultimate goal of this ongoing work is "optimal" sampling in the meta-reasoning sense. Nevertheless, this task is daunting for the following reasons:

- Defining the cost of a sample is not easy, and even if we simply use time-cost as an approximation, we get an intractable meta-reasoning problem.
- Applying the standard myopic and subtree independence assumptions, we run into serious problems. Even in the standard selection problem [7], we get a non-concave utility function and premature stopping of the algorithm. This is due to the fact that the value of information of a single measurement (analogous to a sample in MCTS) is frequently less than its time-cost, even though this is not true for multiple measurements. When applying the selection problem to MCTS, the situation is exacerbated. The utility of an action is usually bounded, and thus in many cases a single sample may be insufficient to change the current best action, *regardless* of its outcome. As a result, we frequently get a *zero* "myopic" value of information for a single sample.

As the ultimate task is extremely difficult to achieve, and even harder to analyze, we begin with simple schemes more amenable to analysis. We then develop a crude value of information (VOI) based scheme. Although the latter is a somewhat ad-hoc attempt to estimate value of information of a sample, it already appears (empirically) to do better than UCT as well as better than all the simpler schemes we propose.

Algorithm 1 Two-stage Monte-Carlo tree search sampling

```
1: procedure ROLLOUT(node, depth=1)
2:     if ISLEAF(node, depth) then
3:         return 0
4:     else
5:         if depth=1 then
6:             action ← FIRSTACTION(node)          ▷ minimizes simple regret, e.g. 1/2-greedy
7:         else
8:             action ← NEXTACTION(node)           ▷ UCB — minimizes cumulative regret
9:         end if
10:        next-node ← NEXTSTATE(node, action)
11:        reward ← TRANSITIONREWARD(node, action, next-node)
12:                + ROLLOUT(next-node, depth+1)   ▷ call ROLLOUT recursively
13:        UPDATESTATS(node, action, reward)
14:    end if
15: end procedure
```

2 Main Results

2.1 Doing better than UCB

1. The ε-greedy allocation scheme has exponential convergence rate of the simple regret $\mathbb{E}r$ (see Appendix B.1.1):

$$\mathbb{E}r_\varepsilon \leq \sum_{j=1}^{K} \Delta_j \left(\frac{n\varepsilon}{K} + \frac{4\sqrt{e}}{\Delta_j^2} \right) e^{-\Delta_j^2 n\varepsilon/8K} \tag{2}$$

2. The UCB allocation scheme exhibits polynomial convergence rate (see Appendix B.1.2):

$$\mathbb{E}r_{ucb} \leq 2 \sum_{j=1}^{K} \Delta_j n^{\frac{-\alpha\Delta_j^2}{8}} \tag{3}$$

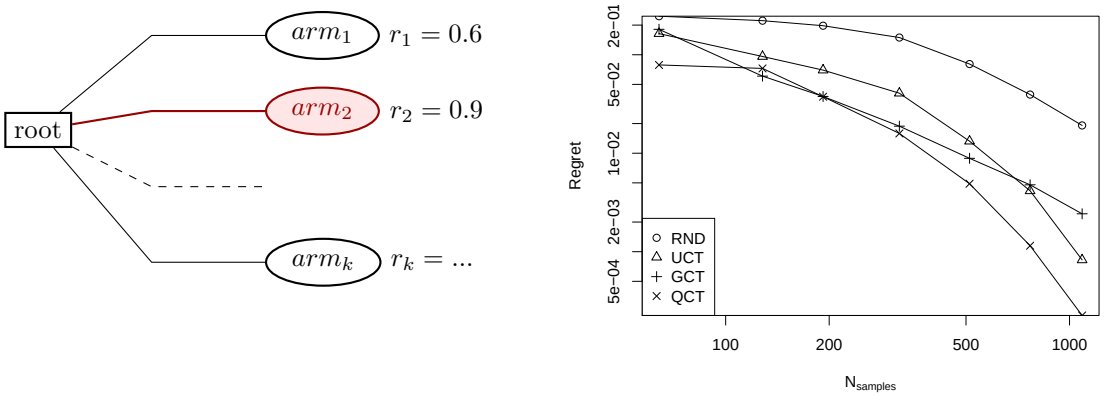
3. A tighter upper regret bound for the ε-greedy allocation scheme depends on ε. The lowest bound is achieved for $\varepsilon = \frac{1}{2}$, and for a large number of arms the bound for the $\frac{1}{2}$-greedy scheme approaches the square of the bound for uniform sampling (see Appendix B.2.1).

4. Another improved allocation scheme for minimizing the simple regret, called **UQB** here, is obtained by substituting $\sqrt{\cdot}$ instead of $\log(\cdot)$ into the formula for UCB and has a superpolynomial convergence rate (see Appendix B.2.2).

2.2 Doing better than UCT

The **two-stage** Monte Carlo tree search sampling scheme selects an action at the current root node according to an algorithm that minimizes the simple regret, such as $\frac{1}{2}$-greedy or UQB, and then selects actions according to UCB. Such two-stage outperforms the UCT sampling scheme that selects actions based on UCB only.

Such sampling scheme is significantly less sensitive to the tuning of the exploration factor (C_p in [5]) of UCT, since the conflict [2] between the need for a larger value of C_p on the first step (simple regret) and a smaller value for the rest of the rollout (cumulative regret) is resolved. In fact, a sampling scheme that uses UCB at all steps but a larger value of C_p for the first step than for the rest of the steps, outperforms UCT. The pseudocode of the two-stage rollout is in Algorithm 1.



a. search tree

b. regret vs. number of samples

Figure 1: Simple regret in MAB

3 Empirical Evaluation

The results were empirically verified on Multi-armed Bandit instances (Section 3.1), on search trees (Section 3.2), and on the sailing domain (Section 3.3), as defined in [5]. The experiments confirmed the hypotheses. The algorithms used in the experiments are:

RND: uniform random samping;

UCT: Upper Confidence Bounds applied to Trees [5];

GCT: UCT with the first step replaced with $\frac{1}{2}$-greedy sampling;

QCT: UCT with the first step replaced with UQB.

3.1 Simple regret in multi-armed bandits

Figure 3.1 presents a comparison of simple regret minimization algorithms for Multi-armed bandits. Figure 3.1.a shows the search tree corresponding to a problem instance. Each arm returns a random reward drawn from a Bernoulli distribution. The search selects an arm and compares the expected reward, unknown to the algorithm during the sampling, to the expected reward of the best arm.

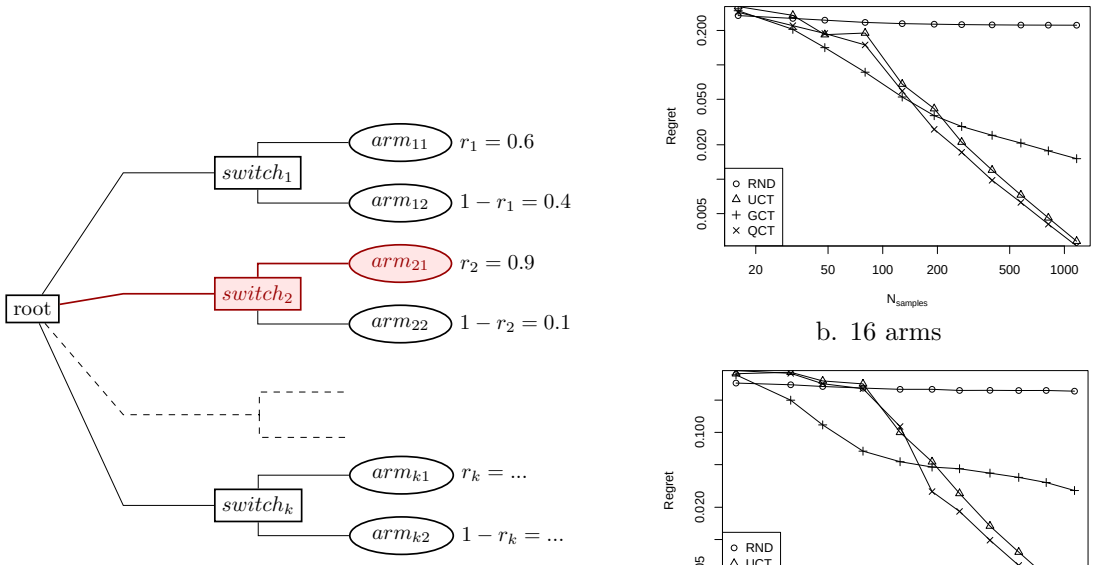
Figure 3.1.b shows the regret vs. the number of samples, averaged over 10^4 experiments for randomly generated instances of 32 arms.

For smaller numbers of samples, $\frac{1}{2}$-greedy achieves the best performance; for large number of samples, QCT outperforms GCT. QCT is better than UCT everywhere except for very small numbers of samples. A combination of GCT and QCT dominates UCT over the whole range.

3.2 Monte Carlo tree search

The second set of experiments was performed on randomly generated trees crafted in such a way that uniform random sampling selects a direction at the root randomly. The degree of the root is a parameter of the tree generator. The degree of all nodes at distance 1 from the root is 2, and all nodes at distance 2 from the roots are leaves. The average reward of two children of each node at distance 1 is 0.5. Thus, a uniform sampling scheme results in the same average reward for all edges at the root, and an adaptive sampling scheme, such as UCT, has to be used.

Figure 3.2 shows a sketch of the search tree (Figure 3.2.a) and the dependency of the regret vs. the number of samples for trees with root degree 16 (Figure 3.2.b) and 64 (Figure 3.2.c). The dependencies



a. search tree

b. 16 arms

c. 64 arms

Figure 2: MCTS: a path to the best arm

look differently from Multi-armed bandit instances, but the algorithms exhibit a similar relative performance: either GCT or QCT is always better than UCT, QCT dominates UCT everywhere except for small numbers of instances. The advantage of QCT grows with the number of arms.

3.3 The sailing domain

Figures 3–6 show results of experiments on the sailing domain. Figure 3 shows the regret vs. the number of samples, computed for a range of values of C_p. Figure 3.a shows the median cost, and Figure 3.b — the minimum costs. UCT is always worse than either $\frac{1}{2}$-greedy (GCT) or QCT, and is sensitive to the value of C_p: the median cost is much higher than the minimum cost for UCT. For both $\frac{1}{2}$-greedy and QCT, the difference is significantly less prominent.

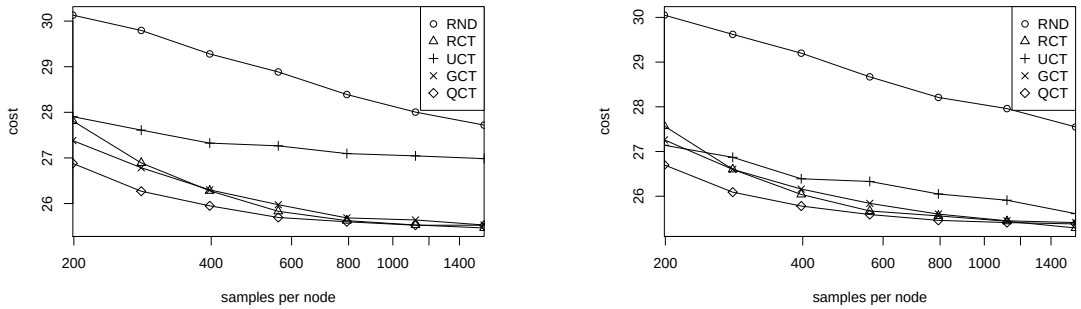
Figure 4 shows the regret vs. the exploration factor for different numbers of samples. QCT is always better than UCT, and $\frac{1}{2}$-greedy is better than UCT expect for a small range of values of the exploration factor C_p.

Figure 5 shows the cost vs. the exploration factor for lakes of different sizes. The relative difference between the sampling schemes becomes more prominent when the lake size increases.

Figure 6 compares QCT with UCT with a different exploration factor at the root (CCT). C_p for the rest of the steps was chosen to ensure the best performance from earlier experiments on the domain. Both algorithms exhibit similar dependency, but as the number of samples grows, QCT achieves smaller average regrets and is less sensitive to the choice of the value for C_p at the first step.

4 Summary and Future Work

The Monte Carlo tree search algorithms presented in the paper differ from UCT at the first step of the rollout, when the 'simple' selection regret is minimized instead of the cumulative regret. Both the



a. median cost b. minimum cost

Figure 3: The sailing domain, 6×6 lake, cost vs. number of rollouts

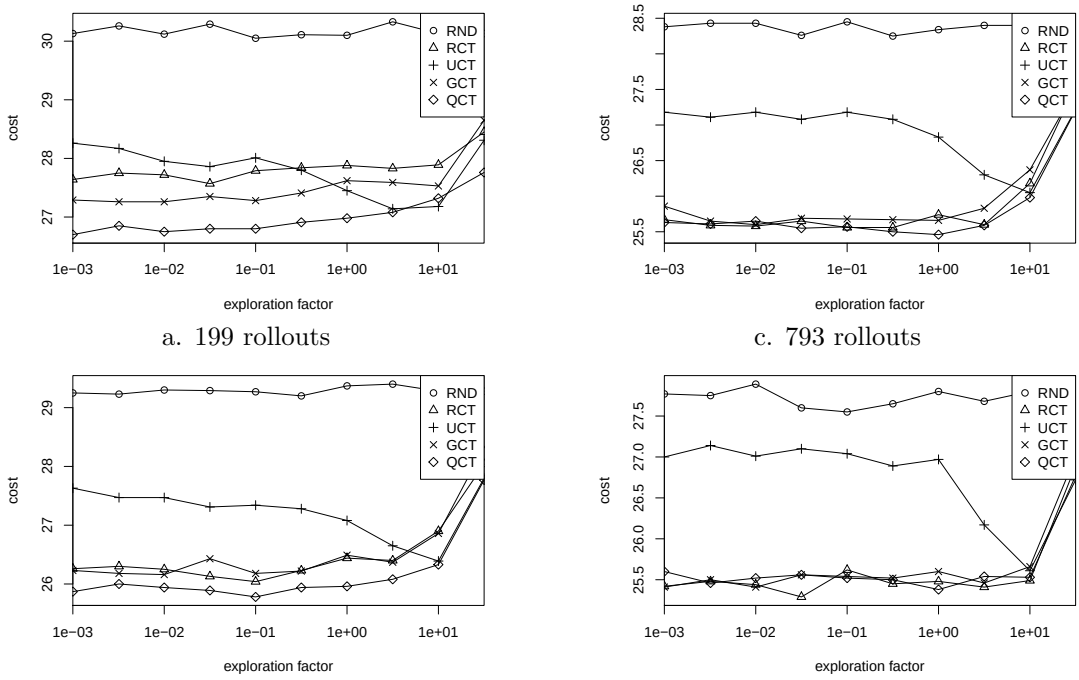
theoretical analysis and the empirical evaluation provide evidence for better general performance of the proposed algorithms.

The improvement is inspired by the notion of value of information (VOI), but VOI is used implicitly in the analysis of the algorithm, rather than computed or learned explicitly in order to plan the rollouts. A further improvement can be achieved by computing or estimating the VOI of the rollouts and choosing a rollout that maximizes the VOI. Using VOI to guide Monte Carlo sampling instead of relying on the sample means of different actions is particularly beneficial when different actions have qualitatively different distributions: for example, when the distribution variance varies significantly between the actions.

VOI of a rollout can be computed when the sample distribution of an action is known up to the parameters, such as the normal distribution with an unknown mean and/or variance. Alternatively, the value of information can be estimated from the set of samples, and the need to assume a particular shape of the distribution can be lifted. In one realization of the latter approach the VOI of performing an action is estimated as *the value of perfect information learned from the outcome of earlier samples of the action divided by the number of samples.* Early experiments with this approach showed the best performance for sets of Bernoulli arms, and even a more prominent improvement for mixed arms (e.g. Bernoulli, triangularly distributed, and fixed arms) (Figure 7): **VCT**, the algorithm which selects an action with the maximum VOI estimate on the first step exhibits lower average regret than other algorithms. Sample-based VOI estimation requires keeping more information than an algorithm based only on the sample mean, but the overhead is small and is justified by the improvement.

Acknowledgments

The research is partially supported by Israel Science Foundation grant 305/09, by the Lynne and William Frankel Center for Computer Sciences, and by the Paul Ivanier Center for Robotics Research and Production Management.



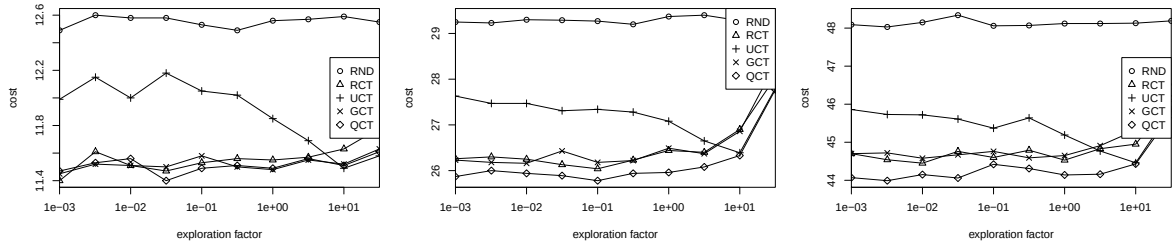
a. 199 rollouts

c. 793 rollouts

b. 397 rollouts

d. 1585 rollouts

Figure 4: The sailing domain, 6×6 lake, cost vs. factor

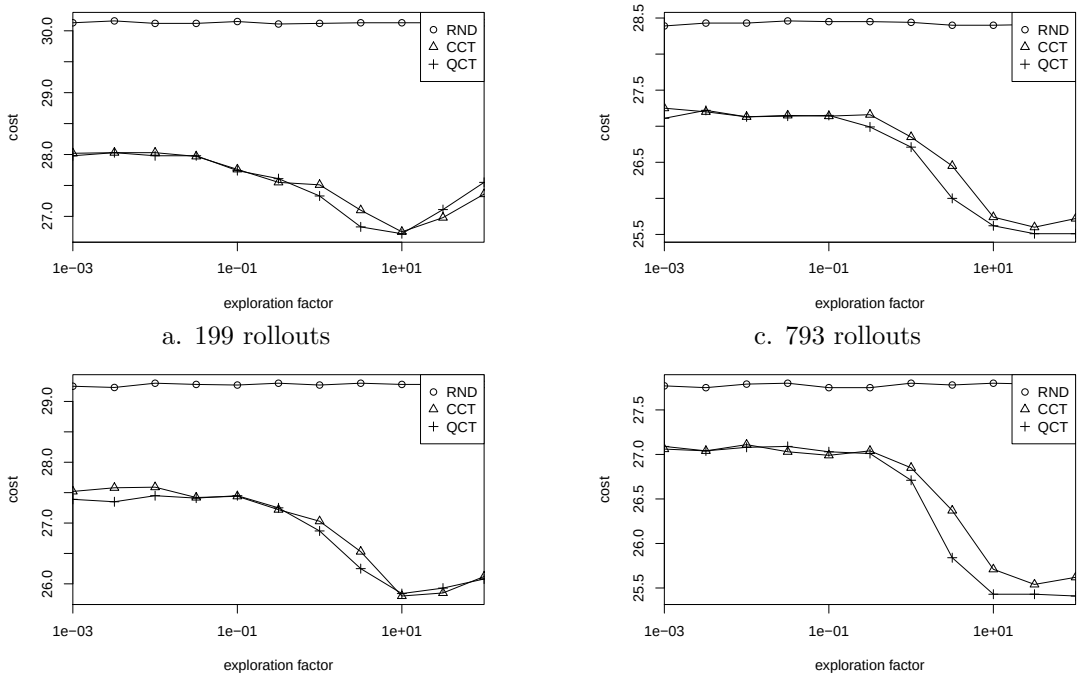


a. 3×3 lake

b. 6×6 lake

b. 10×10 lake

Figure 5: The sailing domain, 397 samples, cost vs. factor



a. 199 rollouts c. 793 rollouts

b. 397 rollouts d. 1585 rollouts

Figure 6: The sailing domain, log vs. sqrt, 6×6 lake

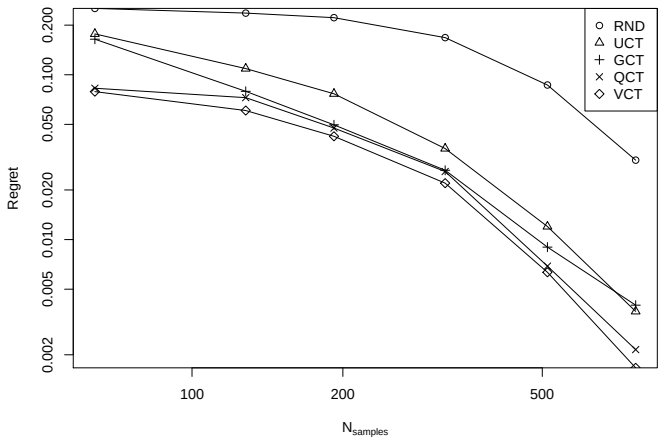


Figure 7: VOI-based action selection (VCT) vs. other algorithms

A Facts

Chernoff bounds (see [4]): for m independent random variables $X_1, X_2, ..., X_m$ taking on values 0 or 1, $X = \sum_{i=1}^{m} X_i$, and $0 \leq \delta \leq 1$:

$$\Pr[X < (1-\delta)\mathbb{E}[X]] < e^{-\delta^2 \mathbb{E}[X]/2} \text{ (Chernoff bound)} \tag{4}$$

Further on, if $\forall i.\mathbb{E}[X_i] = \mu$:

$$\Pr[X < n\mu - a] < e^{-2a^2/n} \text{ (Hoeffding-Chernoff bound)} \tag{5}$$

B Derivations

B.1 Finite-time regret bounds

For an analysis of pure exploration of uniform and UCB algorithms, see also [2].

B.1.1 ε-greedy

For $0 < \varepsilon \leq 1 - \frac{1}{K}$ and $x_0 > 0$ (see Section 3, Proofs, in [1]):

$$\begin{aligned} \Pr(\overline{X}_j = \max_i \overline{X}_i) &\leq 2\left(x_0 \cdot \Pr\left\{T_j^R(n) \leq x_0\right\} + \frac{2}{\Delta_j^2} e^{-\Delta_j^2 \lfloor x_0 \rfloor/2}\right) \\ &\leq 2\left(x_0 \cdot \Pr\left\{T_j^R(n) \leq x_0\right\} + \frac{2 \cdot e^{\Delta_j^2/2}}{\Delta_j^2} e^{-\Delta_j^2 x_0/2}\right) \end{aligned} \tag{6}$$

Number of times ξ the jth arm is selected *randomly*:

$$\xi \triangleq \mathbb{E}\left[T_j^R(n)\right] = \frac{n\varepsilon}{K} \tag{7}$$

By choosing $x_0 = \frac{\xi}{2}$ and applying the Chernoff bound (4), obtain:

$$\begin{aligned} \Pr(\overline{X}_j = \max_i \overline{X}_i) &\leq 2\left(\frac{\xi}{2} e^{-\xi/8} + \frac{2e^{\Delta_j^2/2}}{\Delta_j^2} e^{-\Delta_j^2 \xi/4}\right) \\ &\leq \left(\xi + \frac{4\sqrt{e}}{\Delta_j^2}\right) e^{-\Delta_j^2 \xi/8} \end{aligned} \tag{8}$$

A bound on the simple regret r_ε of an ε-greedy policy follows from substituting (8) into (1):

$$\begin{aligned} \mathbb{E}r_\varepsilon &\leq \sum_{j=1}^{K} \Delta_j \left(\xi + \frac{4\sqrt{e}}{\Delta_j^2}\right) e^{-\Delta_j^2 \xi/8} \\ &\leq K\left(\xi + \frac{4\sqrt{e}}{\Delta_{\min}}\right) e^{-\Delta_{\min}^2 \xi/8} \end{aligned} \tag{9}$$

B.1.2 UCB(α)

n_* pulls of the best arm and n_j pulls of the jth arm are achieved simultaneously when either

$$\sqrt{\frac{\alpha \log n}{n_j - 1}} \geq 1 + \sqrt{\frac{\alpha \log n}{n_*}} \tag{10}$$

or

$$\sqrt{\frac{\alpha \log n}{n_j}} \leq 1 + \sqrt{\frac{\alpha \log n}{n_* - 1}} \tag{11}$$

Inequality (11) can be used to derive a lower bound on the number of pulls of a non-optimal arm, and, consequently, the upper bound on the probability of selecting a non-optimal arm:

$$\begin{aligned} \frac{\alpha \log n}{n_j} &\leq 1 + 2\sqrt{\frac{\alpha \log n}{n_* - 1}} + \frac{\alpha \log n}{n_* - 1} \\ n_j &\geq \frac{\alpha \log n}{1 + 2\sqrt{\frac{\alpha \log n}{n_* - 1}} + \frac{\alpha \log n}{n_* - 1}} \\ &\geq \frac{\alpha \log n}{4} \text{ for } \alpha \log n \leq n_* - 1 \end{aligned} \tag{12}$$

The probability that the jth arm will be chosen can be bounded using the Chernoff-Hoeffding bound:

$$P_j \leq 2 \exp\left(-\frac{\alpha \Delta_j^2 \log n}{8}\right) = 2n^{\frac{-\alpha \Delta_j^2}{8}} \tag{13}$$

The expected simple regret of an UCB(α) algorithm is thus bounded by the following inequality:

$$\mathbb{E} r_{ucb} \leq 2 \sum_{j=1}^{K} \Delta_j n^{\frac{-\alpha \Delta_j^2}{8}} \tag{14}$$

[2] establishes for UCB(α) a similar bound, polynomial in the number of pulls n.

B.2 Asymptotic regret analysis

Bounds (9, 14) were derived ignoring dependency of the allocation on expected values of non-optimal arms. Derivation of better bounds seems complicated. Instead, one can analyze the asymptotic behavior of each of the algorithms under the assumption that the sample mean is close to the expectation.

B.2.1 ε-greedy

A common technique [1] to bound the probability of selecting a non-optimal arm j is to split the interval between μ_j and μ_* at the middle and bound the probability of selecting the jth arm by the sum of the probabilities that the sample mean $\overline{X}_j$ of the jth arm is greater than $\mu_j + \frac{\Delta_j}{2}$ and the sample mean $\overline{X}_*$ of the optimal arm is less than $\mu_* - \frac{\Delta_j}{2}$. Each of the probabilities can be bound using the Chernoff-Hoefding bound.

Splitting the interval between μ_j and μ_* half-way is an arbitrary decision. By selecting a different split point depending on the allocation of samples, one can obtain a tighter bound. On the other hand, there is a value of ε that minimizes the bound for the ε-greedy allocation scheme. Thus, the ε-greedy allocation scheme can be tuned by finding ε that minimizes the bound obtained by selecting the best split point.

Split the interval $[\mu_j, \mu_*]$ at $\mu_j + \delta_j$. The probability P_j for selecting the jth arm is bounded, using the union bound and the Chernoff-Hoeffding bound, as:

$$\begin{aligned} P_j &\leq \Pr[\overline{X}_j > \mu_j + \delta_j] + \Pr[\overline{X}_* < \mu_* - (\Delta_j - \delta_j)] \\ &\leq \exp\left(-2\delta_j^2 n_j\right) + \exp\left(-2(\Delta_j - \delta_j)^2 n_*\right) \end{aligned} \tag{15}$$

For the ε-greedy scheme, $n_i = \frac{\varepsilon n}{K}$, $n_* = (1-\varepsilon)n$, thus

$$P_j \leq \exp\left(-\frac{2\delta_j^2 \varepsilon n}{K-1}\right) + \exp\left(-2(\Delta_j - \delta_j)^2 (1-\varepsilon) n\right) \tag{16}$$

In general, ε and δ which minimize P_j will depend of n. For a constant ε, require that the exponents in both terms are equal

$$\exp\left(-\frac{2\delta_j^2 \varepsilon n}{K-1}\right) = \exp\left(-2(\Delta_j - \delta_j)^2 (1-\varepsilon) n\right) \tag{17}$$

$$\frac{\delta_j}{\Delta_j - \delta_j} = \sqrt{\frac{(K-1)(1-\varepsilon)}{\varepsilon}} \tag{18}$$

and find conditional minima of $P_j(\delta_j, \epsilon)$ satisfying either $\frac{\partial P_j}{\partial \delta_j} = 0$ or $\frac{\partial P_j}{\partial \varepsilon} = 0$.

a. $\frac{\partial P_j}{\partial \delta_j} = 0$

$$\begin{aligned} \frac{4\delta_j \varepsilon n}{K-1} &- 4(\Delta_j - \delta_j)(1-\varepsilon) n = 0 \\ \frac{\delta_j}{\Delta_j - \delta_j} &= \frac{(K-1)(1-\varepsilon)}{\varepsilon} \end{aligned}$$

and together with (18) this gives:

$$\begin{aligned} \delta_j &= \frac{\Delta_j}{2} \\ \varepsilon &= 1 - \frac{1}{K} \\ \mathbb{E} r_{\varepsilon = 1 - 1/K} &\leq B_{uniform} = 2\exp\left(-\frac{\Delta_j^2 n}{2K}\right) \end{aligned} \tag{19}$$

That is, the case $\frac{\partial P_j}{\partial \delta_j} = 0$ corresponds to *uniform random sampling*, and the tightest bound is obtained by splitting the interval at the middle.

b. $\frac{\partial P_j}{\partial \varepsilon} = 0$

$$\begin{aligned} \frac{2\delta_j^2 n}{K-1} &- 2(\Delta_j - \delta_j)^2 n = 0 \\ \frac{\delta_j}{\Delta_j - \delta_j} &= \sqrt{K-1} \end{aligned}$$

and together with (18) this gives:

$$\begin{aligned}
\frac{\delta_j}{\Delta_j} &= \frac{1}{\sqrt{K-1}+1} \\
\varepsilon &= \frac{1}{2} \\
\mathbb{E}r_{\varepsilon=1/2} &\leq B_{1/2\text{-}greedy} = 2\exp\left(-\frac{\Delta_j^2 n}{(\sqrt{K-1}+1)^2}\right) \\
\lim_{K\to\infty} \mathbb{B}r^K_{1/2\text{-}greedy} &= 2\exp\left(-\Delta_j^2 n\right) \\
\text{for large } K\text{: } \mathbb{B}r_{1/2\text{-}greedy} &\approx 2\exp\left(-\frac{\Delta_j^2 n}{K}\right) \qquad (20)
\end{aligned}$$

That is, the case $\frac{\partial P_j}{\partial \varepsilon} = 0$ corresponds to $\frac{1}{2}$-*greedy sampling*, and for a large number of arms K

$$\mathbb{B}r_{1/2\text{-}greedy} \approx \mathbb{B}r^2_{uniform} \qquad (21)$$

Equation (21) suggests that *for a large number of arms* $\frac{1}{2}$-greedy sampling has a much higher convergence rate than uniform sampling.

B.2.2 UCB(α)

The $UCB(\alpha)$ allocation scheme distributes arm pulls such that the upper bound on the probability of selecting a non-optimal arm is asymptotically the same for all arms. Indeed, for large n

$$\begin{aligned}
n_i &\ll n \qquad (22) \\
n_* &\approx n \qquad (23) \\
\sqrt{\frac{2\log n}{n_j}} &\approx \Delta_j + \sqrt{\frac{2\log n}{n_*}} \qquad (24) \\
&\Rightarrow n_j \approx \frac{2\log n}{\Delta_j^2} \qquad (25)
\end{aligned}$$

and by Hoeffding-Chernoff bound the upper bound $\mathbb{B}P_j$ on the probability P_j^{ucb} to choose the jth arm is

$$P_j^{ucb} \leq \mathbb{B}P_j^{ucb} \approx 2\exp\left(-2\Delta_j^2 \frac{2\log n}{\Delta_j^2}\right) = 2n^{-4} \qquad (26)$$

That is, UCB(α) supposedly distributes pulls of non-optimal arms better than ε-greedy scheme which may undersample arms with high expectation. The polynomial, rather than exponential, convergence rate (Equation 14) is due to oversampling the best arm.

Replacing the logarithm with a faster growing subpolynomial function, like square root (UQB), results in a superpolynomial convergence rate of the upper bound:

$$P_j^{uqb} \leq \mathbb{B}P_j^{uqb} \approx 2\exp\left(-\beta\Delta_j^2 \frac{\sqrt{n}}{\Delta_j^2}\right) = 2\exp(-\beta\sqrt{n}) \qquad (27)$$

where β is a constant that depends only on reward bounds. Finite time convergence can still be shown similarly to UCB.

References

[1] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47:235–256, May 2002.

[2] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in finitely-armed and continuous-armed bandits. *Theor. Comput. Sci.*, 412(19):1832–1852, 2011.

[3] Patrick Eyerich, Thomas Keller, Malte Helmert, and Albert ludwigs-universitt Freiburg. High-quality policies for the canadian travelers problem. In *In Proc. AAAI 2010*, pages 51–58, 2010.

[4] Torben Hagerup and C. Rüb. A guided tour of Chernoff bounds. *Inf. Process. Lett.*, 33:305–308, February 1990.

[5] Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *ECML*, pages 282–293, 2006.

[6] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2003.

[7] David Tolpin and Solomon Eyal Shimony. Semi-myopic measurement selection for optimization under uncertainty. Technical Report 10-01, Lynne and William Frankel Center for Computer Science at Ben Gurion University of the Negev, Israel, 2010.