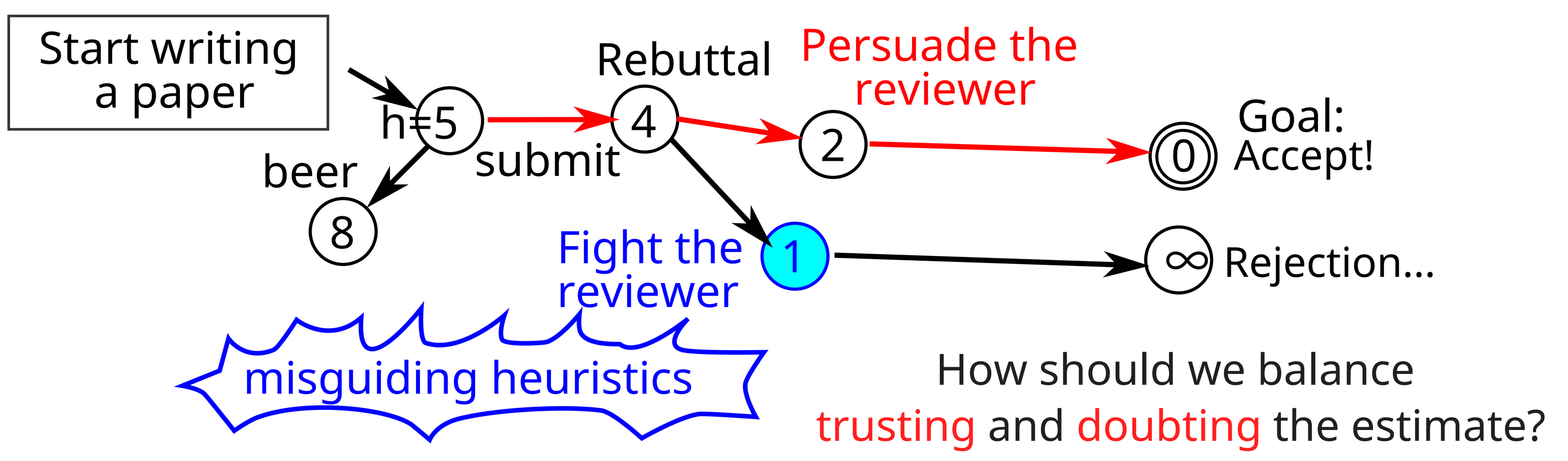
Extreme Value Monte Carlo Tree Search (Ext. Abs)

Masataro Asai, MIT-IBM Watson AI Lab
Stephen Wissow, University of New Hampshire

Background: What's wrong

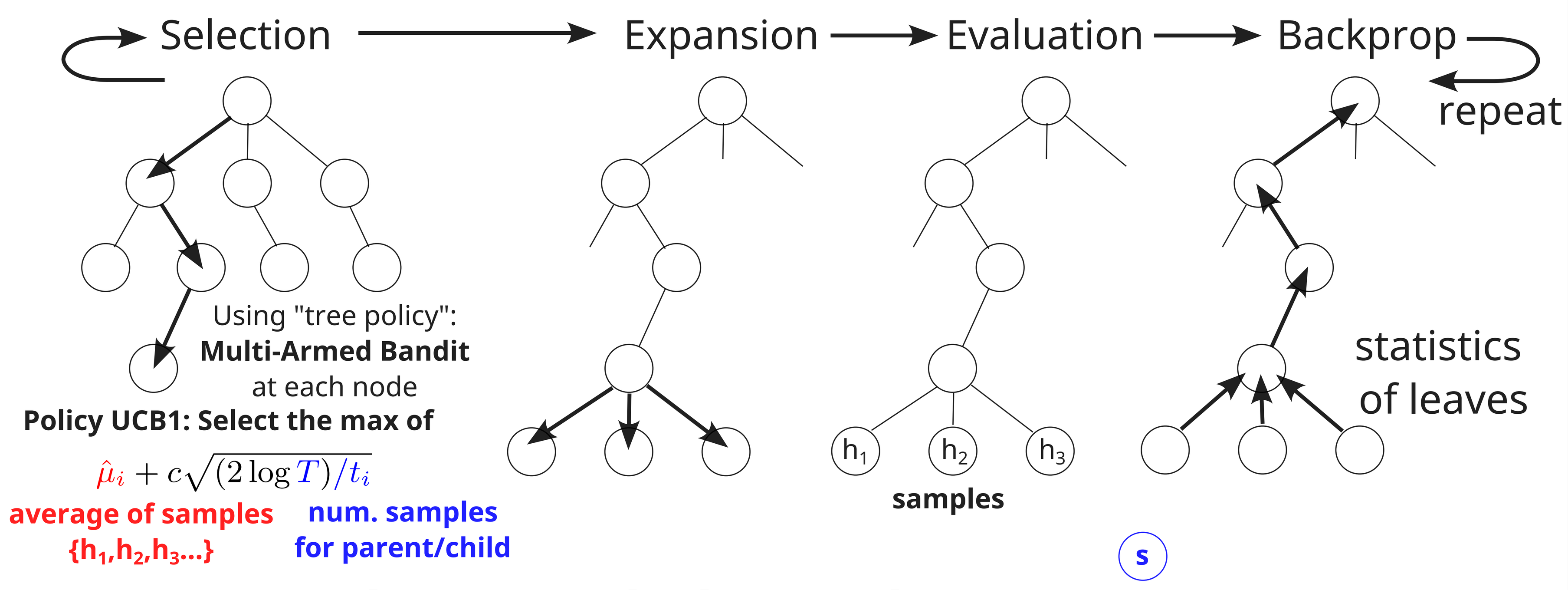
1. Balancing Exploitation vs Exploration

h : an estimate of distance/cost to the goal



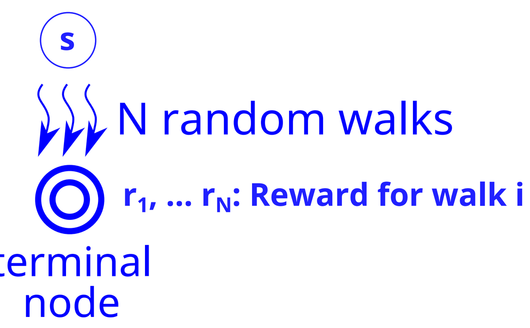
How should we balance trusting and doubting the estimate?

2. Monte-Carlo Tree Search (Kocsis & Szepesvári, 2006)



3. How to evaluate? (Schulte et al, 2014)

For PDDL, domain-independent heuristics makes sense



$h(s) = FF(s)$ $\quad h(s) = \frac{1}{N}\sum r_i$

	Heuristics	Monte-Carlo Simulation
Games	✗	✔
(define (...) ...) Planning	✔	✗

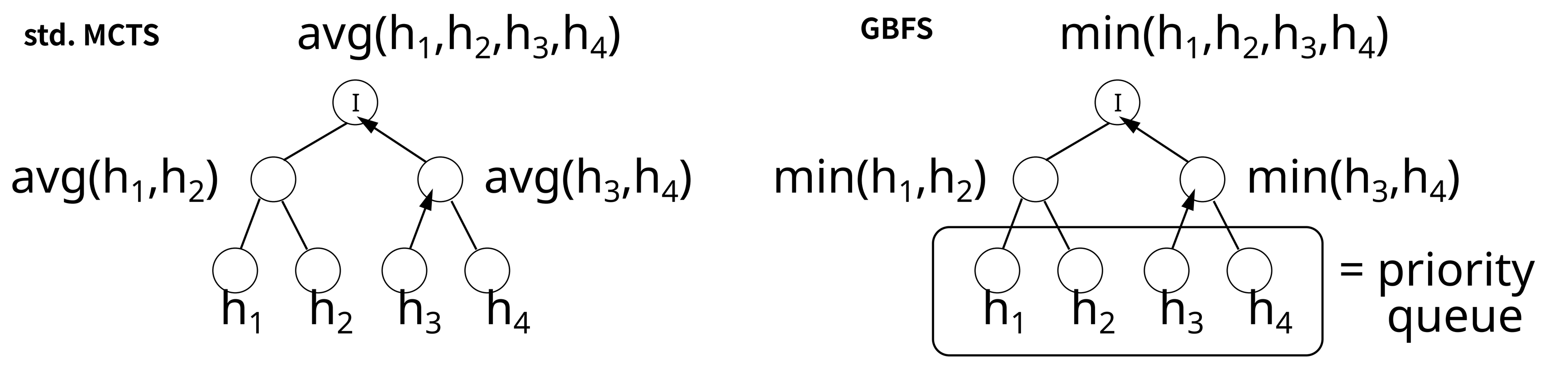
guaranteed to terminate in a fixed number of steps

difficult to design domain-dependent

Infeasible in planning (terminal nodes, a.k.a. goals, are rare!)

4. GBFS is just "min backprop" MCTS! (Schulte et al, 2014)

So, let's call it THTS! (Trial-based Heuristic Tree Search)



5. UCB1 is WRONG (Wissow & Asai, 2023)

UCB1 : designed for 0/1 rewards (games are like that! : win = 1, loss = 0)
BUT Heuristics have no upper bound !!! ([0, 1] is an overspecification)

Algorithm	Assumption	
UCB1	*known finite support* distributions, like [0, 1]	
UCB1-Normal	Gaussian + assumptions	(may not hold)
UCB1-Normal2	**Gaussian + different assumptions**	(more likely to hold in classical planning)

Solution:

- Use an **unbounded distribution**! Gaussian: [-∞, ∞]
- Backpropagates both (mean, variance): N(μ, σ)
- As seen here, quite powerful vs UCB1-based MCTS in classical planning

6. Gaussian is better but is STILL WRONG (this paper)

Using the average is SO wrong

- **Gaussian:** no bounds at all — $h \notin [-\infty, \infty] = R$
  Heuristics have unknown bounds !!! — $h \in [0, \infty]$
  (underspecification) — $h_{add} \in [h_{max}, \infty]$, lmcut $\in [0, h^+]$ ....
- **We are interested in good nodes. Why do we use the average?**
  - Average takes ***ALL bad nodes*** into consideration
- **Why change it when it's not broken?** GBFS uses the minimum.
  - Schulte et.al. proposed min-backup (full-bellman backup), but it is not good
  - Why not good? → min-backup lacks statistical justifications. (bandit theory)
  - **What's the statistical theory of minimum/maximum (extremum)?**
- **Dead-ends (h=∞) break the average.** Average of [2,3,∞,7,5] is ∞.
  - Existing work removes the dead-ends because otherwise it doesn't work
  - GBFS implicitly does it (**minimum** can discard ∞ naturally)
  - **"Removing them just to make the algorithm work"** is **ad-hoc and wrong**

Extreme Value Theory: What's right

- **Statistical theory of the maximum (≠ average)**
- **Used in safety-critical applications: e.g. Maximum water level**
- **There are two types:**
  - ***Method of block maxima*:** (block)
    e.g. Predict next *monthly* maximum from several *monthly* maxima
  - ***Peaks-over-Threshold*: Predict the exceedance over the threshold**

What to model?	Limit Theorem (N→∞)	converges to
Average	Central Limit Theorem	Gaussian Distribution
Block maxima	Fisher–Tippett–Gnedenko	Extreme Value Distribution (EVD)
Exceedance	Pickands–Balkema–de Haan	Generalized Pareto (GP) Distribution

$$\mathrm{GP}(\mathrm{x} \mid \theta, \sigma, \xi) = \begin{cases} \frac{1}{\sigma}\left(1+\xi\frac{\mathrm{x}-\theta}{\sigma}\right)^{-\frac{\xi+1}{\xi}} & (\xi \neq 0) \\ \frac{1}{\sigma}\exp\left(-\frac{\mathrm{x}-\theta}{\sigma}\right) & (\xi = 0) \end{cases} \quad (\mathrm{x} > \theta)$$

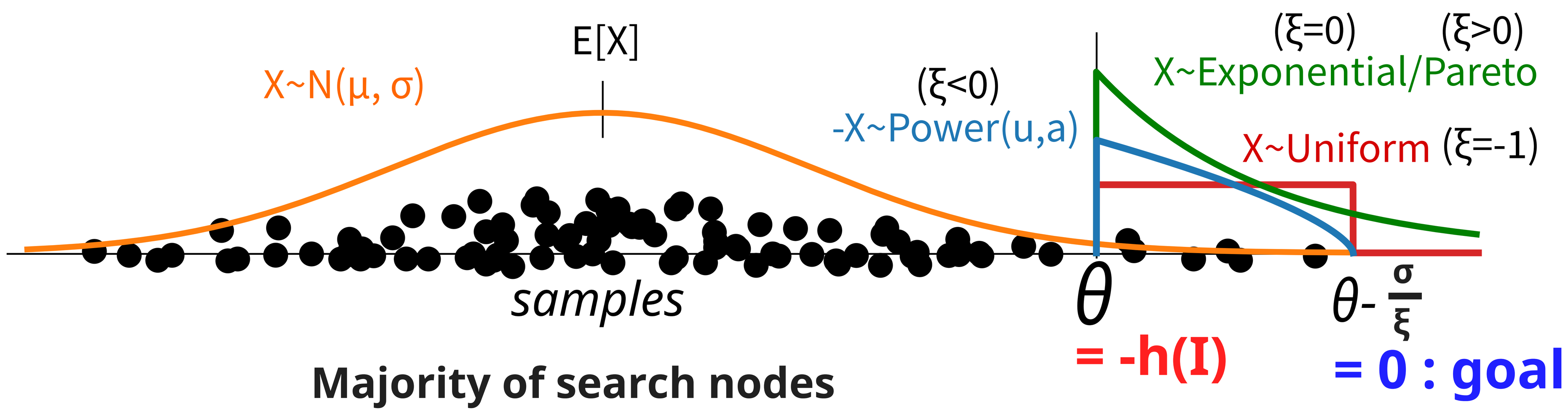
Note: It models the maximum, so we should invert the signs for minimization

Theoretical reason for removing the dead-ends (x = -∞)

- **We predict the exceedance above θ = -h(I) Initial heuristic value**

fit N(μ, σ) = based on all samples (incl. bad h), compute the average & the variance

fit GP(θ, σ, ξ) = discard samples below the threshold θ, compute the maximum and shape ξ (similarity to high water mark/benches)



Majority of search nodes in the entire state space are far from the goal, thus are useless / must be discarded

GP = distribution near the goal !!!

Backprop GP, not Gaussian

We focus on GP's special cases: Uniform and Power. (see paper for why)

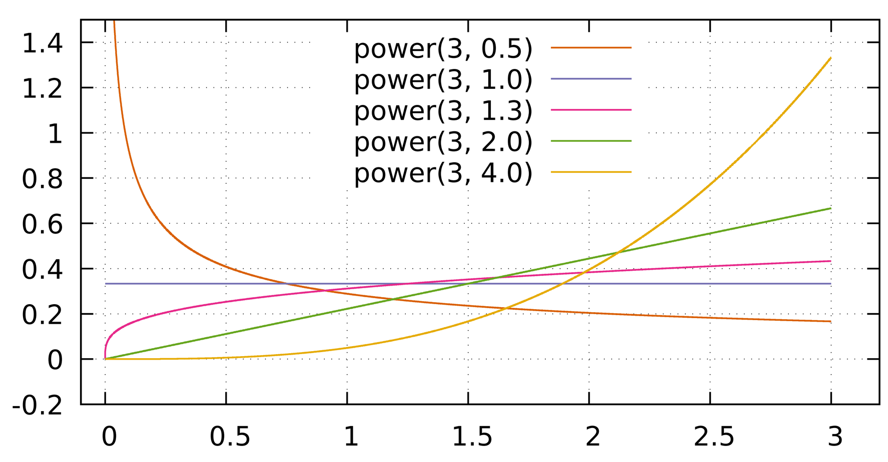
(GP with ξ<0) $$\mathrm{Pow}(\mathrm{x}|u,a) = \frac{a x^{a-1}}{u^a}. \quad (0 < x < u,\ 0 < a)$$

(GP with ξ=-1) $$U(\mathrm{x}|l,u) = \frac{1}{u-l}. \quad (l < x < u)$$

Backpropagate min h_i, max h_i, mean log h_i.

Power: $\hat{u} = \max_i x_i$ *and* $\hat{a} = \left(\log\hat{u} - \frac{1}{N}\sum_i \log x_i\right)^{-1}$ **x_i: heuristic of leaf i**

Uniform: $\hat{u} = \max_i x_i$ *and* $\hat{l} = \min_i x_i$



Power's shape parameter *a* : Rarity of h near 0

***a* is estimated from backpropagaged heuristics**
Small *a* ⇔ nodes with small h are common
Large *a* ⇔ nodes with small h are rare

Multi-Armed Bandit for GP (with regret bounds!)

$$\mathrm{LCB1\text{-}Uniform}_i = \frac{\hat{u}_i+\hat{l}_i}{2} - (\hat{u}_i - \hat{l}_i)\sqrt{6t_i \log T}$$
$$\mathrm{LCB1\text{-}Power}_i = \frac{\hat{u}_i\hat{a}_i}{\hat{a}_i+1} - \hat{u}_i\sqrt{6t_i \log T}$$

LCB1-Uniform/Power have worst-case polynomial, best-case constant regrets. Let $\alpha \in [0,1]$ be an unknown problem-dependent constant and u_i, l_i, a_i be unknown ground-truth parameters of Uniform and Power distributions of arm i. The regret is respectively bounded as follows, where $\beta = (2-\alpha)^{1/a_i}$.

$$\frac{24(u_i-l_i)^2(1-\alpha)^2\log T}{\Delta_i^2} + 1 + 2C + \frac{(1-\alpha)T(T+1)(2T+1)}{3}$$
$$\frac{2u_i^2(3-\beta)^2(\beta-1)^2\log T}{\Delta_i^2} + 1 + 2C + \frac{(1-\alpha)T(T+1)(2T+1)}{3}$$

Results

Num. solved on 24 IPC domains w/ 10^4 evaluations

$h =$	h^{FF}	h^{add}	h^{max}	h^{GC}	h^{FF}+PO	h^{FF}+DE	h^{FF}+DE+PO
GBFS	538	518	224	354	-	489	-
Softmin-Type(h)	576	542.6	297.2	357.6	-	578	-
GUCT **Uses UCB1**	412	397.8	228.4	285.2	454	389.2	439.4
-Normal **Uses UCB1-Normal**	283.4	265	212	233.4	372.4	289	381.6
*-Normal **+ backprop min**	318.8	300	215.2	246.2	378.05	304.4	386.7
-Normal2	581.8	535.8	316.6	379	621	518	578
*-Normal2	567.2	533.8	263	341	618	511.4	567.8
-Power	**596**	541.8	**450.6**	463.2	**623.4**	413.6	**583**
-Uniform	**594.8**	**543.8**	**450.6**	**463.8**	**626.4**	416.4	**583**

FD/C++ implementation is on the way and showing promising results