Scale-Adaptive Balancing of Exploration and Exploitation in Classical Planning

Stephen Wissow and Masataro Asai

University of New Hampshire

MIT-IBM Watson AI Lab

This work was supported through DTIC contract FA8075-18-D-0008, Task Order FA807520F0060, Task 4 - Autonomous Defensive Cyber Operations (DCO) Research & Development (R&D).

Agile unit-cost planning

- **agile search**: minimize planning time, ignore solution cost

Agile unit-cost planning

- **agile search**: minimize planning time, ignore solution cost
- "best-first": always an estimate—when to trust it?

Agile unit-cost planning

- **agile search**: minimize planning time, ignore solution cost
- "best-first": always an estimate—when to trust it?
- need distributional belief about distance-to-go

Agile unit-cost planning

- **agile search**: minimize planning time, ignore solution cost
- "best-first": always an estimate—when to trust it?
- need distributional belief about distance-to-go
- previous work has considered distributional heuristics but without complete statistical analysis:

Agile unit-cost planning

- **agile search**: minimize planning time, ignore solution cost
- "best-first": always an estimate—when to trust it?
- need distributional belief about distance-to-go
- previous work has considered distributional heuristics but without complete statistical analysis:
  - Ethan Burns (diss., UNH 2013), O'Ceallaigh & Ruml (SoCS-15), Mitchell et al. (AAAI-19), Fickert, Gu, & Ruml (IJCAI-21)

Agile unit-cost planning

- **agile search**: minimize planning time, ignore solution cost
- "best-first": always an estimate—when to trust it?
- need distributional belief about distance-to-go
- previous work has considered distributional heuristics but without complete statistical analysis:
  - Ethan Burns (diss., UNH 2013), O'Ceallaigh & Ruml (SoCS-15), Mitchell et al. (AAAI-19), Fickert, Gu, & Ruml (IJCAI-21)
- this work:

Agile unit-cost planning

- **agile search**: minimize planning time, ignore solution cost
- "best-first": always an estimate—when to trust it?
- need distributional belief about distance-to-go
- previous work has considered distributional heuristics but without complete statistical analysis:
  - Ethan Burns (diss., UNH 2013), O'Ceallaigh & Ruml (SoCS-15), Mitchell et al. (AAAI-19), Fickert, Gu, & Ruml (IJCAI-21)
- this work:
  - first principled adaptation of Monte Carlo Tree Search (MCTS) for forward state space search

Agile unit-cost planning

- **agile search**: minimize planning time, ignore solution cost
- "best-first": always an estimate—when to trust it?
- need distributional belief about distance-to-go
- previous work has considered distributional heuristics but without complete statistical analysis:
  - Ethan Burns (diss., UNH 2013), O'Ceallaigh & Ruml (SoCS-15), Mitchell et al. (AAAI-19), Fickert, Gu, & Ruml (IJCAI-21)
- this work:
  - first principled adaptation of Monte Carlo Tree Search (MCTS) for forward state space search
  - can readily extend to non-unit cost

Agile unit-cost planning

- **agile search**: minimize planning time, ignore solution cost
- "best-first": always an estimate—when to trust it?
- need distributional belief about distance-to-go
- previous work has considered distributional heuristics but without complete statistical analysis:
  - Ethan Burns (diss., UNH 2013), O'Ceallaigh & Ruml (SoCS-15), Mitchell et al. (AAAI-19), Fickert, Gu, & Ruml (IJCAI-21)
- this work:
  - first principled adaptation of Monte Carlo Tree Search (MCTS) for forward state space search
  - can readily extend to non-unit cost
  - empirically better than GBFS without ad hoc tweaks

Trial-Based Heuristic Tree Search (THTS)

THTS (Schulte & Keller SoCS-14) first applied MCTS to planning

Trial-Based Heuristic Tree Search (THTS)

THTS (Schulte & Keller SoCS-14) first applied MCTS to planning

in each iteration:

1. **selection**
2. **expansion**
3. **evaluation**
4. **backup**

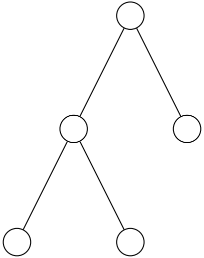
Trial-Based Heuristic Tree Search (THTS)

THTS (Schulte & Keller SoCS-14) first applied MCTS to planning

in each iteration:

1. **selection**
2. **expansion**
3. **evaluation**
4. **backup**

partial tree, mid-search:



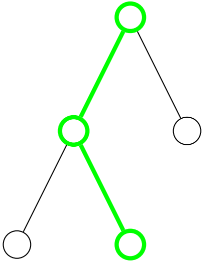
Trial-Based Heuristic Tree Search (THTS)

THTS (Schulte & Keller SoCS-14) first applied MCTS to planning

in each iteration:

1. **selection**
2. **expansion**
3. **evaluation**
4. **backup**

partial tree, mid-search:



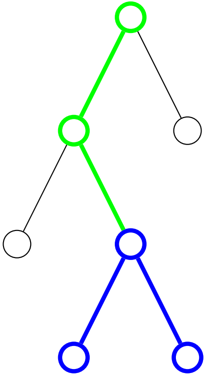
Trial-Based Heuristic Tree Search (THTS)

THTS (Schulte & Keller SoCS-14) first applied MCTS to planning

in each iteration:

1. **selection**
2. **expansion**
3. **evaluation**
4. **backup**

partial tree, mid-search:



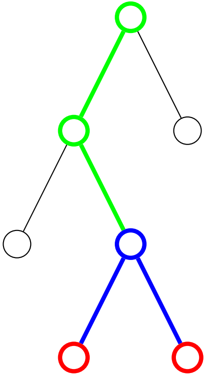
Trial-Based Heuristic Tree Search (THTS)

THTS (Schulte & Keller SoCS-14) first applied MCTS to planning

in each iteration:

1. **selection**
2. **expansion**
3. **evaluation**
4. **backup**

partial tree, mid-search:



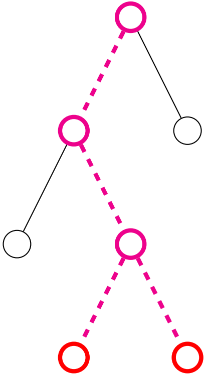
Trial-Based Heuristic Tree Search (THTS)

THTS (Schulte & Keller SoCS-14) first applied MCTS to planning

in each iteration:

1. **selection**
2. **expansion**
3. **evaluation**
4. **backup**

partial tree, mid-search:



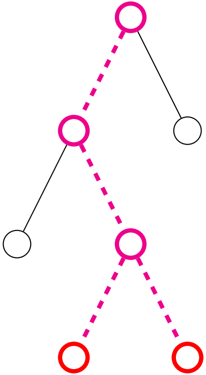
Trial-Based Heuristic Tree Search (THTS)

THTS (Schulte & Keller SoCS-14) first applied MCTS to planning

in each iteration:

1. **selection**
2. **expansion**
3. **evaluation**
4. **backup**

partial tree, mid-search:



nodes store things like:

h heuristic value

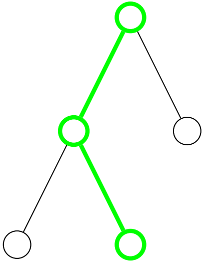
$\hat{\mu}$ sample mean

t visitation counter

...modified via backup

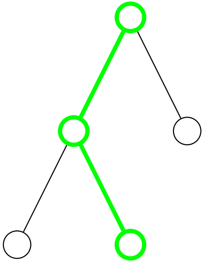
Multi-Armed Bandit (MAB)

- MAB for **selection** determines MCTS behavior



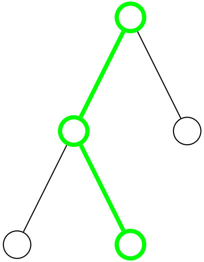
Multi-Armed Bandit (MAB)

- MAB for **selection** determines MCTS behavior
- **balance exploration and exploitation**



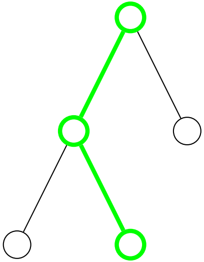
Multi-Armed Bandit (MAB)

- MAB for **selection** determines MCTS behavior
- **balance exploration and exploitation**
- theoretically rigorous



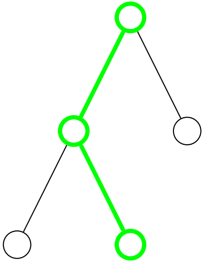
Multi-Armed Bandit (MAB)

- MAB for **selection** determines MCTS behavior
- **balance exploration and exploitation**
- theoretically rigorous
- UCB1 (Auer et al. ML-02)



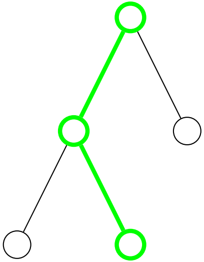
Multi-Armed Bandit (MAB)

- MAB for **selection** determines MCTS behavior
- **balance exploration and exploitation**
- theoretically rigorous
- UCB1 (Auer et al. ML-02)
  - used with MCTS for games: UCT (Kocsis and Szepesvári ECML-06)



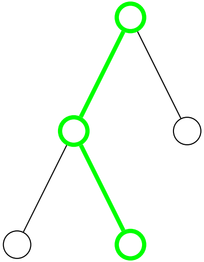
Multi-Armed Bandit (MAB)

- MAB for **selection** determines MCTS behavior
- **balance exploration and exploitation**
- theoretically rigorous
- UCB1 (Auer et al. ML-02)
  - used with MCTS for games: UCT (Kocsis and Szepesvári ECML-06)
  - rewards $\{0, 1\}$ or $\{-1, 1\}$



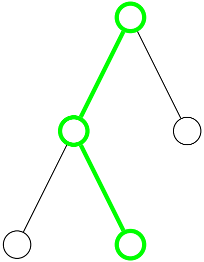
Multi-Armed Bandit (MAB)

- MAB for **selection** determines MCTS behavior
- **balance exploration and exploitation**
- theoretically rigorous
- UCB1 (Auer et al. ML-02)
  - used with MCTS for games: UCT (Kocsis and Szepesvári ECML-06)
  - rewards $\{0, 1\}$ or $\{-1, 1\}$
  - average win rate: Bernoulli distribution



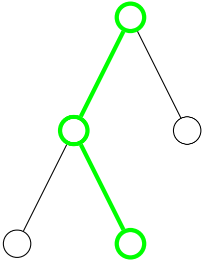
Multi-Armed Bandit (MAB)

- MAB for **selection** determines MCTS behavior
- **balance exploration and exploitation**
- theoretically rigorous
- UCB1 (Auer et al. ML-02)
  - used with MCTS for games: UCT (Kocsis and Szepesvári ECML-06)
  - rewards $\{0, 1\}$ or $\{-1, 1\}$
  - average win rate: Bernoulli distribution
- but **heuristics lack a priori known upper bounds**

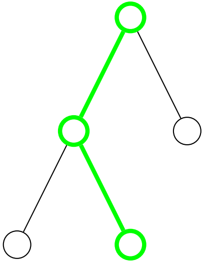


Multi-Armed Bandit (MAB)

- MAB for **selection** determines MCTS behavior
- **balance exploration and exploitation**
- theoretically rigorous
- UCB1 (Auer et al. ML-02)
  - used with MCTS for games: UCT (Kocsis and Szepesvári ECML-06)
  - rewards $\{0, 1\}$ or $\{-1, 1\}$
  - average win rate: Bernoulli distribution
- but **heuristics lack a priori known upper bounds**
- need *unbounded* distribution, e.g. Gaussian

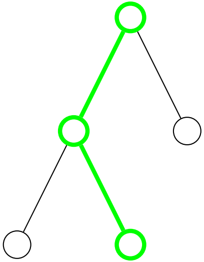


Multi-Armed Bandit (MAB)



- MAB for **selection** determines MCTS behavior
- **balance exploration and exploitation**
- theoretically rigorous
- UCB1 (Auer et al. ML-02)
  - used with MCTS for games: UCT (Kocsis and Szepesvári ECML-06)
  - rewards $\{0,1\}$ or $\{-1,1\}$
  - average win rate: Bernoulli distribution
- but **heuristics lack a priori known upper bounds**
- need *unbounded* distribution, e.g. Gaussian
- to measure uncertainty: consider variance σ^2, not just mean μ

Multi-Armed Bandit (MAB)



- MAB for **selection** determines MCTS behavior
- **balance exploration and exploitation**
- theoretically rigorous
- UCB1 (Auer et al. ML-02)
  - used with MCTS for games: UCT (Kocsis and Szepesvári ECML-06)
  - rewards $\{0, 1\}$ or $\{-1, 1\}$
  - average win rate: Bernoulli distribution
- but **heuristics lack a priori known upper bounds**
- need *unbounded* distribution, e.g. Gaussian
- to measure uncertainty: consider variance σ^2, not just mean μ
- some existing MABs use μ, σ^2, but with drawbacks

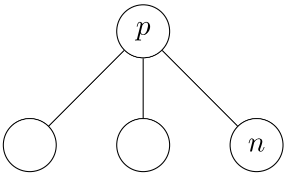
GUCT-01[1] (Schulte & Keller SoCS-14)

- GUCT-01 adapts UCB1 by normalizing siblings to $[0, 1]$

[1](renamed from original Greedy-UCT)

GUCT-01[1] (Schulte & Keller SoCS-14)

- GUCT-01 adapts UCB1 by normalizing siblings to $[0,1]$



$$M = \max_{n' \in S(p)} h(n')$$

$$m = \min_{n' \in S(p)} h(n')$$

$$f_{\mathrm{GUCT-01}}(n) = \overset{\textbf{exploitation}}{\frac{h(n) - m}{M - m}} - \overset{\textbf{exploration}}{{\color{red}c}\sqrt{(2\log T)/t}}$$

$\boldsymbol{p}$ parent of n
$\boldsymbol{S(p)}$ successors of p

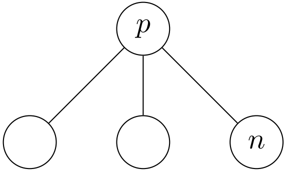
$\boldsymbol{T}$ visitation counter at p
$\boldsymbol{t}$ visitation counter at n

$\boldsymbol{c}$ exploration coefficient

[1] (renamed from original Greedy-UCT)

GUCT-01[1] (Schulte & Keller SoCS-14)

- GUCT-01 adapts UCB1 by normalizing siblings to $[0, 1]$



$$M = \max_{n' \in S(p)} h(n')$$

$$m = \min_{n' \in S(p)} h(n')$$

$$f_{\mathrm{GUCT-01}}(n) = \overbrace{\frac{h(n) - m}{M - m}}^{\textbf{exploitation}} - \overbrace{c\sqrt{(2 \log T)/t}}^{\textbf{exploration}}$$

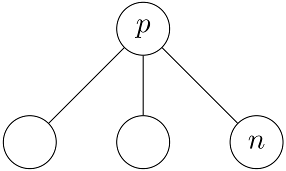
$\boldsymbol{p}$ parent of n
$\boldsymbol{S(p)}$ successors of p
$\boldsymbol{T}$ visitation counter at p
$\boldsymbol{t}$ visitation counter at n
$\boldsymbol{c}$ exploration coefficient

- UCB1 assumes all arms share same bounded reward distribution

[1] (renamed from original Greedy-UCT)

GUCT-01[1] (Schulte & Keller SoCS-14)

- GUCT-01 adapts UCB1 by normalizing siblings to $[0, 1]$



$$M = \max_{n' \in S(p)} h(n')$$

$$m = \min_{n' \in S(p)} h(n')$$

$$f_{\mathrm{GUCT-01}}(n) = \underbrace{\frac{h(n) - m}{M - m}}_{\textbf{exploitation}} - \underbrace{c\sqrt{(2 \log T)/t}}_{\textbf{exploration}}$$

$\boldsymbol{p}$ parent of n

$\boldsymbol{S(p)}$ successors of p

$\boldsymbol{T}$ visitation counter at p

$\boldsymbol{t}$ visitation counter at n

$\boldsymbol{c}$ exploration coefficient

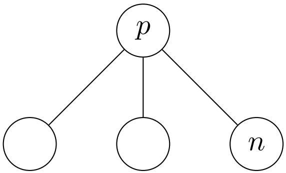
- UCB1 assumes all arms share same bounded reward distribution
- but h range varies per subtree!

[1](renamed from original Greedy-UCT)

$M - m$

Factor out normalization:

- preserve node ordering
- demonstrate outsize impact on exploration term: will explore too much



$$M = \max_{n' \in S(p)} h(n')$$

$$m = \min_{n' \in S(p)} h(n')$$

$$\begin{array}{rcc} & \textbf{exploitation} & \textbf{exploration} \\ f_{\text{GUCT}-01}(n) = & \dfrac{h(n) - m}{M - m} & - c\sqrt{(2\log T)/t} \\ m + (M - m) f_{\text{GUCT}-01}(n) = & h(n) & - c(M - m)\sqrt{(2\log T)/t} \end{array}$$

GUCT-Normal2: theoretically sound for classical planning

$$\begin{array}{lcc} & \text{exploitation} & \text{exploration} \\ f_{\text{GUCT-01}}(n) = & \dfrac{h(n)-m}{M-m} & -c\sqrt{(2\log T)/t} \\ f_{\text{GUCT-Normal2}}(n) = & \color{red}{h(n)} & -\color{red}{\hat{\sigma}_n}\sqrt{2\log T/\color{red}{1}} \end{array}$$

(differences in red)

GUCT-Normal2: theoretically sound for classical planning

$$
\begin{array}{rcc}
 & \text{exploitation} & \text{exploration} \\
f_{\mathrm{GUCT-01}}(n) = & \dfrac{h(n)-m}{M-m} & -c\sqrt{(2\log T)/t} \\
f_{\mathrm{GUCT-Normal2}}(n) = & \color{red}{h(n)} & -\color{red}{\hat{\sigma}_n}\sqrt{2\log T/\color{red}{1}}
\end{array}
$$

(differences in red)

GUCT-Normal2:

GUCT-Normal2: theoretically sound for classical planning

$$
\begin{array}{lll}
 & \text{exploitation} & \text{exploration} \\
f_{\mathrm{GUCT-01}}(n) = & \dfrac{h(n) - m}{M - m} & -c\sqrt{(2\log T)/t} \\
f_{\mathrm{GUCT-Normal2}}(n) = & \color{red}{h(n)} & -\color{red}{\hat{\sigma}_n}\sqrt{2\log T/\color{red}{1}}
\end{array}
$$

(differences in red)

GUCT-Normal2:

- no longer normalize $h(n)$ to artificial bound

GUCT-Normal2: theoretically sound for classical planning

$$
\begin{array}{rcc}
 & \text{exploitation} & \text{exploration} \\
f_{\mathrm{GUCT-01}}(n) = & \dfrac{h(n)-m}{M-m} & -c\sqrt{(2\log T)/t} \\
f_{\mathrm{GUCT-Normal2}}(n) = & \color{red}{h(n)} & -\color{red}{\hat{\sigma}_n}\sqrt{2\log T/\color{red}{1}}
\end{array}
$$

(differences in red)

GUCT-Normal2:

- no longer normalize $h(n)$ to artificial bound
- exploration coefficient $\hat{\sigma}_n$ learned online

GUCT-Normal2: theoretically sound for classical planning

$$\begin{array}{rcc} & \text{exploitation} & \text{exploration} \\ f_{\mathrm{GUCT-01}}(n) = & \dfrac{h(n) - m}{M - m} & -c\sqrt{(2\log T)/t} \\ f_{\mathrm{GUCT-Normal2}}(n) = & \color{red}{h(n)} & -\color{red}{\hat{\sigma}_n}\sqrt{2\log T/\color{red}{1}} \end{array}$$

(differences in red)

GUCT-Normal2:

- no longer normalize $h(n)$ to artificial bound
- exploration coefficient $\hat{\sigma}_n$ learned online
  - data: h in leaf nodes

GUCT-Normal2: theoretically sound for classical planning

$$
\begin{array}{rcc}
 & \text{exploitation} & \text{exploration} \\
f_{\mathrm{GUCT-01}}(n) = & \dfrac{h(n) - m}{M - m} & -c\sqrt{(2\log T)/t} \\
f_{\mathrm{GUCT-Normal2}}(n) = & h(n) & -\hat{\sigma}_n\sqrt{2\log T/1}
\end{array}
$$

(differences in red)

GUCT-Normal2:

- no longer normalize $h(n)$ to artificial bound
- exploration coefficient $\hat{\sigma}_n$ learned online
  - data: h in leaf nodes
- $\hat{\sigma}_n$ specific to n

GUCT-Normal2: theoretically sound for classical planning

$$\begin{array}{rcl} & \text{exploitation} & \text{exploration} \\ f_{\mathrm{GUCT-01}}(n) = & \dfrac{h(n)-m}{M-m} & -c\sqrt{(2\log T)/t} \\ f_{\mathrm{GUCT-Normal2}}(n) = & h(n) & -\hat{\sigma}_n\sqrt{2\log T/1} \end{array}$$

(differences in red)

GUCT-Normal2:

- no longer normalize $h(n)$ to artificial bound
- exploration coefficient $\hat{\sigma}_n$ learned online
  - data: h in leaf nodes
- $\hat{\sigma}_n$ specific to n
  - won't explore too much (compared to $M-m$)

GUCT-Normal2: theoretically sound for classical planning

	exploitation	exploration
$f_{\mathrm{GUCT-01}}(n) =$	$\dfrac{h(n) - m}{M - m}$	$-c\sqrt{(2\log T)/t}$
$f_{\mathrm{GUCT-Normal2}}(n) =$	$h(n)$	$-\hat{\sigma}_n\sqrt{2\log T/1}$

(differences in red)

GUCT-Normal2:

- no longer normalize $h(n)$ to artificial bound
- exploration coefficient $\hat{\sigma}_n$ learned online
  - data: h in leaf nodes
- $\hat{\sigma}_n$ specific to n
  - won't explore too much (compared to $M - m$)
  - adapts exploration to each child's subtree

Better IPC coverage under 10k node evaluations

h	GBFS	GUCT-01 (c)	**GUCT-Normal2**
h^{FF}			
h^{add}			
$h^{\max}$			
h^{GC} (for Atari)			

Better IPC coverage under 10k node evaluations

h	GBFS	GUCT-01 (c)	**GUCT-Normal2**
h^{FF}	522.4		
h^{add}	501.6		
$h^{\max}$	221.4		
h^{GC} (for Atari)	351.2		

Better IPC coverage under 10k node evaluations

h	GBFS	GUCT-01 (c)	**GUCT-Normal2**
h^{FF}	522.4	369.6 (0.5) 354.8 (1.0)	
h^{add}	501.6	345.2 (0.5) 312.8 (1.0)	
h^{max}	221.4	242.2 (0.5) 227.6 (1.0)	
h^{GC} (for Atari)	351.2	307.0 (0.5) 295.2 (1.0)	

Better IPC coverage under 10k node evaluations

h	GBFS	GUCT-01 (c)	**GUCT-Normal2**
h^{FF}	522.4	369.6 (0.5) 354.8 (1.0)	**563.8**
h^{add}	501.6	345.2 (0.5) 312.8 (1.0)	**519.2**
h^{max}	221.4	242.2 (0.5) 227.6 (1.0)	**301.0**
h^{GC} (for Atari)	351.2	307.0 (0.5) 295.2 (1.0)	**374.6**

Better IPC coverage under 10k node evaluations

h	GBFS	GUCT-01 (c)	**GUCT-Normal2**
h^{FF}	522.4	369.6 (0.5) 354.8 (1.0)	**563.8**
h^{add}	501.6	345.2 (0.5) 312.8 (1.0)	**519.2**
$h^{\max}$	221.4	242.2 (0.5) 227.6 (1.0)	**301.0**
h^{GC} (for Atari)	351.2	307.0 (0.5) 295.2 (1.0)	**374.6**

h^{FF}+PO

h^{FF}+DE

h^{FF}+DE+PO

Better IPC coverage under 10k node evaluations

h	GBFS	GUCT-01 (c)	**GUCT-Normal2**
h^{FF}	522.4	369.6 (0.5) 354.8 (1.0)	**563.8**
h^{add}	501.6	345.2 (0.5) 312.8 (1.0)	**519.2**
$h^{\max}$	221.4	242.2 (0.5) 227.6 (1.0)	**301.0**
h^{GC} (for Atari)	351.2	307.0 (0.5) 295.2 (1.0)	**374.6**
h^{FF}+PO	-		
h^{FF}+DE	474.0		
h^{FF}+DE+PO	-		

Better IPC coverage under 10k node evaluations

h	GBFS	GUCT-01 (c)	**GUCT-Normal2**
h^{FF}	522.4	369.6 (0.5) 354.8 (1.0)	**563.8**
h^{add}	501.6	345.2 (0.5) 312.8 (1.0)	**519.2**
$h^{\max}$	221.4	242.2 (0.5) 227.6 (1.0)	**301.0**
h^{GC} (for Atari)	351.2	307.0 (0.5) 295.2 (1.0)	**374.6**
h^{FF}+PO	-	403.2 (0.5) 387.0 (1.0)	
h^{FF}+DE	474.0	355.6 (0.5) 344.8 (1.0)	
h^{FF}+DE+PO	-	406.4 (0.5) 404.4 (1.0)	

Better IPC coverage under 10k node evaluations

h	GBFS	GUCT-01 (c)	**GUCT-Normal2**
h^{FF}	522.4	369.6 (0.5) 354.8 (1.0)	**563.8**
h^{add}	501.6	345.2 (0.5) 312.8 (1.0)	**519.2**
h^{max}	221.4	242.2 (0.5) 227.6 (1.0)	**301.0**
h^{GC} (for Atari)	351.2	307.0 (0.5) 295.2 (1.0)	**374.6**
h^{FF}+PO	-	403.2 (0.5) 387.0 (1.0)	**596.4**
h^{FF}+DE	474.0	355.6 (0.5) 344.8 (1.0)	**496.8**
h^{FF}+DE+PO	-	406.4 (0.5) 404.4 (1.0)	**550.8**

Conclusion

GUCT-Normal2 is the first to check all the boxes:

Conclusion

GUCT-Normal2 is the first to check all the boxes:

1. reason explicitly about balancing exploration and exploitation

Conclusion

GUCT-Normal2 is the first to check all the boxes:

1. reason explicitly about balancing exploration and exploitation
2. statistically sound

Conclusion

GUCT-Normal2 is the first to check all the boxes:

1. reason explicitly about balancing exploration and exploitation
2. statistically sound
3. we prove its regret bound: read our paper!

Conclusion

GUCT-Normal2 is the first to check all the boxes:

1. reason explicitly about balancing exploration and exploitation
2. statistically sound
3. we prove its regret bound: read our paper!
4. empirical success on wide variety of domains

Conclusion

GUCT-Normal2 is the first to check all the boxes:

1. reason explicitly about balancing exploration and exploitation
2. statistically sound
3. we prove its regret bound: read our paper!
4. empirical success on wide variety of domains
5. performs well with or without ad hoc enhancements

Conclusion

GUCT-Normal2 is the first to check all the boxes:

1. reason explicitly about balancing exploration and exploitation
2. statistically sound
3. we prove its regret bound: read our paper!
4. empirical success on wide variety of domains
5. performs well with or without ad hoc enhancements

Conclusion

GUCT-Normal2 is the first to check all the boxes:

1. reason explicitly about balancing exploration and exploitation
2. statistically sound
3. we prove its regret bound: read our paper!
4. empirical success on wide variety of domains
5. performs well with or without ad hoc enhancements

Future work:

Conclusion

GUCT-Normal2 is the first to check all the boxes:

1. reason explicitly about balancing exploration and exploitation
2. statistically sound
3. we prove its regret bound: read our paper!
4. empirical success on wide variety of domains
5. performs well with or without ad hoc enhancements

Future work:

- CPU time instead of node evaluations; reimplement in Fast Downward (Helmert JAIR-06) instead of Pyperplan (Alkhazraji et al. 2016).

Conclusion

GUCT-Normal2 is the first to check all the boxes:

1. reason explicitly about balancing exploration and exploitation
2. statistically sound
3. we prove its regret bound: read our paper!
4. empirical success on wide variety of domains
5. performs well with or without ad hoc enhancements

Future work:

- CPU time instead of node evaluations; reimplement in Fast Downward (Helmert JAIR-06) instead of Pyperplan (Alkhazraji et al. 2016).
- consider distributions other than Gaussian, with yet greater fidelity to heuristic search setting

Conclusion

GUCT-Normal2 is the first to check all the boxes:

1. reason explicitly about balancing exploration and exploitation
2. statistically sound
3. we prove its regret bound: read our paper!
4. empirical success on wide variety of domains
5. performs well with or without ad hoc enhancements

Future work:

- CPU time instead of node evaluations; reimplement in Fast Downward (Helmert JAIR-06) instead of Pyperplan (Alkhazraji et al. 2016).
- consider distributions other than Gaussian, with yet greater fidelity to heuristic search setting
- non-unit cost setting

More on S&K

- empirical: only improved over GBFS with preferred operators (PO), deferred heuristic evaluation (DE)
- regret bounds: GUCT-Normal2's polynomial regret bound expected to be better than GUCT-01's logarithmic regret bound in practice in context of deterministic, discrete, finite-state space search like planning, **where estimated variance can approach or equal true variance**.

Full Coverage Results

$h =$	h^{FF}		h^{add}		$h^{\max}$		h^{GC}		h^{FF}+PO		h^{FF}+DE		h^{FF}+DE+PO	
$c =$	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1
GUCT	413.2	396.4	405.8	373.8	224.8	222.2	296	278	439.2	411.8	418.6	354.6	450	393.2
*	508.8	440.8	496.2	453.8	239.4	234.2	306.2	303	542.4	448	441.8	386.8	477	422
-01	369.6	354.8	345.2	312.8	242.2	227.6	307	295.2	403.2	387	355.6	344.8	406.4	404.4
*-01	393.6	372	373	343.6	236.2	226.4	306.2	289.8	430.2	401.2	377.6	363	426.2	421.2
-V	329.8	307.2	325	297.6	215	200	264.8	243.8	383.8	348.4	334.4	310	384.4	377.4
-Normal	-	278	-	261.4	-	209.2	-	231.8	-	331.6	-	269.2	-	342.6
*-Normal	-	311.6	-	294.8	-	212.2	-	244	-	338.2	-	285.2	-	343.8
-Normal2	-	**563.8**	-	**519.2**	-	**301**	-	**374.6**	-	**596.4**	-	**496.8**	-	**550.8**
*-Normal2	-	551.2	-	516.2	-	258.2	-	338.6	-	593.8	-	490.6	-	543.4
GBFS	-	522.4	-	501.6	-	221.4	-	351.2	-	-	-	474	-	-

10k node evaluation limit, average of 5 runs, no results for configurations unsupported by Pyperplan, subset of IPC benchmarks compatible with PDDL extensions supported by Pyperplan.
Terminated experiments when grounding exceeded 5 min., removing 751 problem instances across 24 domains.

Weaknesses of previous MABs

- UCB1-Normal (Auer et al. ML-02)
  - scale-adaptive, but relies on conjectures that are not guaranteed to hold
- UCB1-Tuned (Auer et al. ML-02):
  - assumes bounded reward distribution
  - lacks regret bound
- UCB-V (Audibert et al. 2009):
  - proven regret bound, but assumes bounded reward distribution
  - needs initialization pulls
- Bayes-UCT2 (Tesauro et al. 2010):
  - lacks regret bound
  - convergence proved only for bounded reward distributions
  - only tested on synthetic trees of fixed depth, width, and rewards