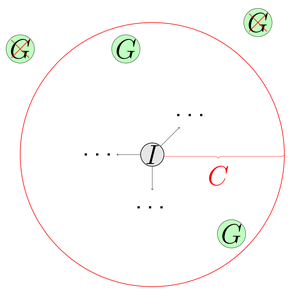
Bounded-Cost Search Using Estimates of Uncertainty

Maximilian Fickert, Tianyi Gu, Wheeler Ruml

SAARLAND UNIVERSITY
SAARBRÜCKEN GRADUATE SCHOOL OF COMPUTER SCIENCE

University of New Hampshire

The Problem Setting



- Motivation: many real-world problems are too hard to solve optimally. Need bounded-cost solution!
- Problem: initial state (I), goal states (G), and a cost bound C.
- Objective: Find a solution with cost at most C as fast as possible.

Previous Approaches

- **Standard heuristic search** with pruning on C
- **Potential Search (PTS)**[1]
  $\rightarrow$ best-first search on $\frac{h}{C-g}$
- **Bounded-cost Explicit Estimation Search (BEES)**[2]
  $\rightarrow$ focal search:
  - *open* sorted by f, only nodes with $g + h \leq C$
  - *focal* sorted by d, only nodes with $g + \hat{h} \leq C$
    ($\hat{h}$ is corrected for the observed heuristic error[3])

[1] Stern, Puzis, and Felner 2011.
[2] Thayer et al. 2012.
[3] Thayer, Dionne, and Ruml 2011.

Expected Effort Search

Best-first search on the expected effort:[4] $\frac{T}{p}$

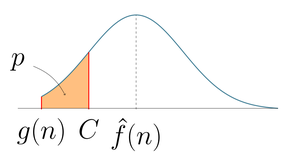
- $T(n)$: search effort to find a solution under n
- $p(n)$: probability that n leads to a solution within C

n_1 $\quad T = 10,\ p = 0.5 \quad \rightsquigarrow 20$

n_2 $\quad T = 6,\ p = 0.25 \quad \rightsquigarrow 24$

How to obtain T and p?

- Obtaining T use distance-to-go d
- Obtaining p from belief distributions



Theoretical Analysis

XES optimizes search effort, assuming

- the search explores one subtree at a time,
- subtrees are independent, and
- subtrees are abandoned after spending $T(n)$ time.

[4] First suggested by Dobson and Haslum (HSDIP'17)

Using p in BEES

BEES*95*:

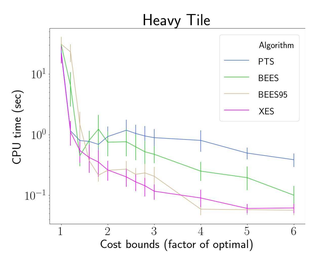
- *open* sorted by f, only nodes with $g + h \leq C$
- *focal* sorted by d, only nodes with ~~$g + \hat{h} \leq C$~~ $p(n) > 95\%$

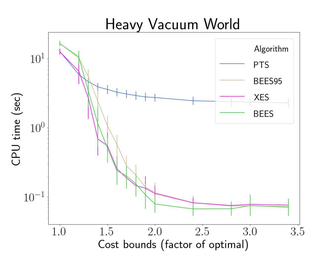
Experiments

Planning Domains: IPC'18 bounded-cost track

Coverage	GBFS	PTS	BEES	BEES95	XES
Agricola (20)	**1**	0	0	0	0
Caldera (20)	8	10	10	12	**13**
Caldera-split (20)	**4**	2	2	2	2
DataNetwork (20)	2	0	3	3	**4**
Nurikabe (20)	4	10	10	**11**	9
Settlers (20)	4	5	10	**11**	**11**
Snake (20)	4	**5**	4	4	**5**
Spider (20)	7	**11**	10	10	9
Termes (20)	11	9	11	11	**13**
Sum (180)	45	52	60	64	**66**
Expansions ($*10^3$)	1.93	3.93	2.10	2.25	**1.77**

Search Domains:





Bounded-cost algorithms dominate GBFS; XES is best overall.