Best-first Utility-guided Search

Wheeler Ruml and Minh B. Do

Embedded Reasoning Area
Palo Alto Research Center
{ruml, minhdo}@parc.com

Overview

A* takes too long: we must trade cost for time.

Overview

- Overview
- Anytime Algs
- BUGSY
- Properties
- Evaluation
- Gridworld
- MSA
- Planning
- Summary

A* takes too long: we must trade cost for time.

$$U(s,t) = -w_f \cdot f(s) - w_t \cdot t$$

Overview

- Overview
- Anytime Algs
- Bugsy
- Properties
- Evaluation
- Gridworld
- MSA
- Planning
- Summary

A* takes too long: we must trade cost for time.

$$U(s,t) = -w_f \cdot f(s) - w_t \cdot t$$

Anytime algorithms are annoying to use and to design.

Utility-guided search is a promising alternative.

The Anytime Approach

The Anytime Approach

- Overview
- Anytime Algs
- Bugsy
- Properties
- Evaluation
- Gridworld
- MSA
- Planning
- Summary

Requires a termination policy, assuming:

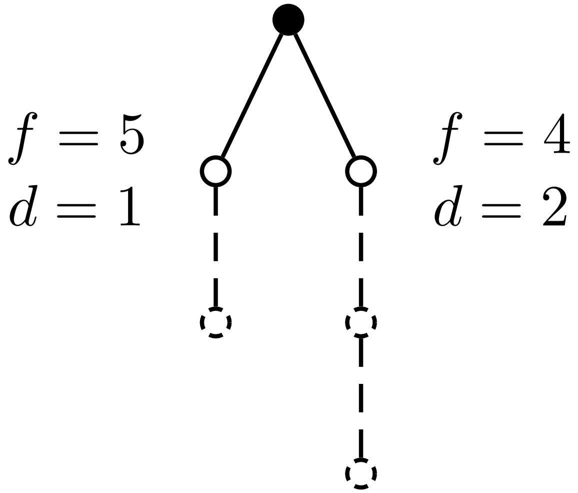
- relevant features for predicting progress are known
- training data available
- new instance is similar in relevant aspects to training
- relevant aspects are known

The Anytime Approach

Requires a termination policy, assuming:

- relevant features for predicting progress are known
- training data available
- new instance is similar in relevant aspects to training
- relevant aspects are known

Impossible to design optimally:



Must know the user's utility function!

Best-first Utility-guided Search, Yes!

- Want best-first search according to:

$$U(n) = \max_{s \text{ under } n} (-w_f \cdot f(s) - w_t \cdot t(s))$$

Best-first Utility-guided Search, Yes!

- Want best-first search according to:

$$U(n) = \max_{s \text{ under } n} (-w_f \cdot f(s) - w_t \cdot t(s))$$

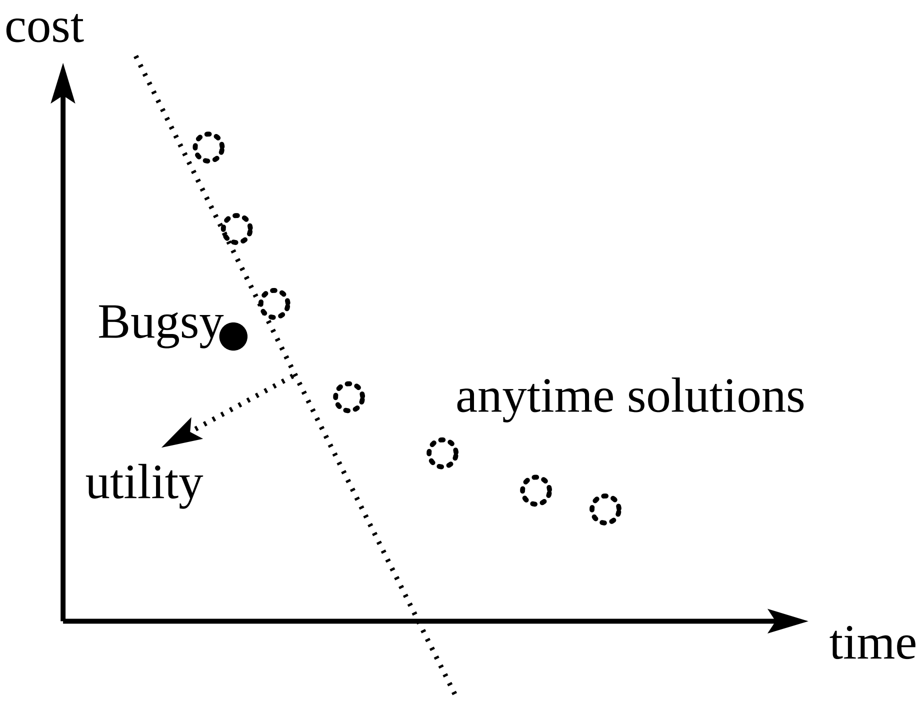
- Approximate s under n by cheapest and nearest
  - $f(\textit{cheapest}) = f(n) = g(n) + h(n)$
  - $f(\textit{nearest})$ seems straightforward in many domains
- Estimate $d(n)$ and convert to $t(n)$.
  - $d(n)$ seems straightforward in many domains

Properties

- Overview
- Anytime Algs
- BUGSY
- Properties
- Evaluation
- Gridworld
- MSA
- Planning
- Summary

Different from anytime algorithms

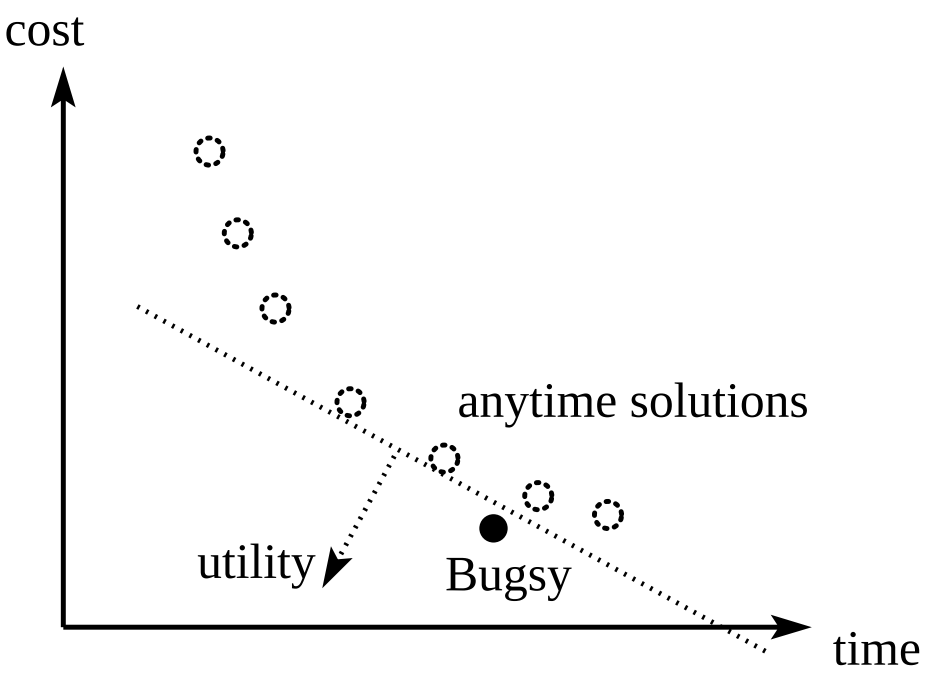
- no need for termination policy, training data
- can spend all effort pursuing one solution
- no fixed trade-off



Properties

Different from anytime algorithms

- no need for termination policy, training data
- can spend all effort pursuing one solution
- no fixed trade-off



Properties

Different from anytime algorithms

- no need for termination policy, training data
- can spend all effort pursuing one solution
- no fixed trade-off

Similar to weighted A* iff $h = d$.

- but intuitive meaning for weight
- otherwise, exploits additional information

Reasonable properties

- Complete if h and d are reasonable
- Optimal if h and d are perfect

Empirical Evaluation

- Algorithms:
  - BUGSY
  - Anytime Replanning A* (ARA*), Likhachev et al. (2004)
  - Anytime A* (AA*), Hansen et al. (1997)
  - Greedy (Gr), Doran and Michie (1966)
  - A*, Hart et al. (1968)
- Wide variety of utility functions.
- Record CPU time and solution quality for every solution.
- Assume clairvoyant termination for anytime algorithms.
- Normalize utilities from 0–100.
- See paper for full results.

(These results are conservative.)

Gridworld Pathfinding

- Overview
- Anytime Algs
- Bugsy
- Properties
- Evaluation
- Gridworld
- MSA
- Planning
- Summary

$U()$	Bugsy	ARA*	AA*	Gr	A*
time only	**100**	**100**	**100**	**100**	59
500 microsec	**100**	99	99	99	59
1 msec	**99**	98	**99**	98	59
5 msec	**99**	91	93	90	59
10 msec	**99**	82	86	80	59
50 msec	**97**	25	54	19	65
0.1 sec	**97**	60	63	19	82
cost only	**98**	**98**	**98**	19	**98**

Multiple Sequence Alignment

$U()$	Bugsy	ARA*	AA*	Gr	A*
time only	**100**	**100**	**100**	**100**	54
0.1 sec	**99**	97	98	96	54
0.5 sec	**92**	83	88	76	52
1 sec	**80**	68	79	54	51
5 sec	**75**	68	71	25	73
10 secs	**78**	75	74	25	**78**
cost only	**82**	**82**	**82**	24	**82**

Temporal Planning

$U()$	Bugsy	ARA*	AA*	A*
	zenotravel-7			
500 microsec	**100**	69	0	81
1 msec	**100**	71	0	83
5 msec	**100**	74	0	85
10 msec	**100**	84	0	96
50 msec	91	91	0	**100**
0.5 sec	97	97	0	**100**
5 sec	99	99	0	**100**
	rovers-5			
500 microsec	**100**	67	0	62
1 msec	**100**	72	0	66
5 msec	**100**	77	0	71
10 msec	92	**100**	0	93
50 msec	78	**100**	0	93

Summary

- Anytime algorithms are annoying to use and to design.
- Utility-based search is a promising and practical alternative.
- Extendable to solving deadlines and plan execution deadlines.
- There's information beyond $g(n)$ and $h(n)$, namely $d(n)$.