RoxyBot:

An Approximately Optimal TAC Agent

Amy Greenwald

Brown University

with

Justin Boyan

NASA Ames Research Center

North Carolina State University

E-Commerce Seminar

November 29, 2000

TAC Market Game

Score

$\boxed{\text{Client Utility - Expenditures}}$

Supply

- Flights Inbound and Outbound
- Hotels Grand Hotel and Le FleaBag Inn
- Entertainment Red Sox, Symphony, Phantom

Auctions

- Flights infinite supply, prices follow random walk, clear continuously, no resale permitted
- Hotels ascending, multi-unit, 16th price auctions, transactions clear at auction close (early closings after random period of inactivity), no resale
- Entertainment continuous double auctions, initial endowment, resale is permitted

TAC Market Game

Demand

Client	IAD	IDD	HV	BRS	SY	PH
1	1	3	99	134	118	65
2	1	4	131	170	47	49
3	1	2	147	13	55	49
4	3	4	145	130	60	85
5	1	4	82	136	68	87
6	2	4	53	94	51	105
7	1	3	54	156	126	71
8	1	5	113	119	187	143

Feasible Packages

- arrival date prior to departure date
- same hotel on all intermediate nights
- at most one entertainment event per night
- at most one of each type of entertainment

TAC Market Game

Utility

$$\text{utility} = 1000 - \text{travelPenalty} + \text{hotelBonus} + \text{funBonus}$$

$$\begin{aligned}
\text{travelPenalty} &= 100(|\text{IAD} - \text{AD}| + |\text{IDD} - \text{DD}|) \\
\text{hotelBonus} &= \begin{cases} \text{HV} & \text{if H} = \text{G} \\ 0 & \text{otherwise} \end{cases} \\
\text{funBonus} &= \text{entertainment values}
\end{aligned}$$

Allocation

Client	AD	DD	H	Ticket	Utility
1	1	3	G	SY1, BRS2	1351
2	1	3	G	BRS1	1201
3	1	2	G	—	1147
4	3	4	G	BRS3	1275
5	1	3	F	BRS1, PH2	1123
6	3	4	G	PH3	1058
7	1	3	F	SY1, BRS2	1282
8	1	5	G	PH1, SY3, BRS4	1562

Key TAC Features

Substitutes $u(A) + u(B) \geq u(A \cup B)$

Complements $u(A) + u(B) \leq u(A \cup B)$

Simultaneous (not Combinatorial) **Auctions**

Naive Strategy

BID UP TO THE VALUE OF GOOD x

Examples

- Red Sox or symphony for $100, but not both
- Inbound Flight day 1, outbound flight day 2, and Grand Hotel for $1000

Key TAC Challenges

Impossible: what is the value of good x?

Possible: what is the value of a set of goods X?

RoxyBot's Key Decision

GIVEN PRICES OF TAC GOODS, HOW MANY COPIES OF EACH IS DESIRED?

Completion: find optimal completion given current holdings and prices

Allocation: find optimal allocation of goods to clients

Optimal vs. Greedy

Example
Clients in Turn

	BRS	SY	PH
1	90	80	70
2	175	150	125

Utility $[1 \leftarrow \text{BRS}, 2 \leftarrow \text{SY}] = 240$
Utility $[1 \leftarrow \text{SY}, 2 \leftarrow \text{BRS}] = 255$

Example
Travel and Entertainment Packages in Turn

RoxyBot's Architecture

(A) REPEAT

1. Update current prices and holdings
2. Estimate clearing prices and build pricelines
3. Run completer to find optimal buy/sell quantities
4. Set bid/ask prices strategically

   UNTIL game over

(B) Run allocator to compute optimal allocation

Combinatorial Auctions

TAC Allocation Problem $\cong$ Multi-Unit Winner Determination Problem

Multi-Set of goods G

- n_j copies of good $j \in G$

Set of bids B, with $b \in B$

- b is a price-quantity pair (p, q), with $p \in \mathbb{R}^+$
- q is a vector of quantities $(q_j)_{j \in G}$

Allocation $A \subseteq B$
Feasible Allocation $F \subseteq B$

$$\forall j, \sum_{(p,(q_j)) \in F} q_j \leq n_j$$

Winner Determination = Revenue Maximization

$$\max_{F \subseteq B} \sum_{(p,(q_j)) \in F} p$$

Theorem [Rothkopf, *et. al.*, 1998]
Winner Determination is NP-Complete

Combinatorial Auctions

dumbed-down version of

TAC Completion Problem

$\cong$ Multi-Unit Winner Determination Problem with Reserve Prices

Multi-Set of goods G

- n_j copies of good $j \in G$
- reserve price r_j for good $j \in G$

Set of bids B, with $b \in B$

- b is a price-quantity pair (p, q), with $p \in \mathbb{R}^+$
- q is a vector of quantities $(q_j)_{j \in G}$

Allocation $A \subseteq B$

Feasible Allocation $F \subseteq B$

$$\forall j, \sum_{(p,(q_j)) \in F} q_j \leq n_j$$

Winner Determination = Revenue Maximization

$$\max_{F \subseteq B} \sum_{(p,(q_j)) \in F} (p - \vec{r} \cdot \vec{q})$$

Combinatorial Auctions

TAC Completion Problem $\cong$

Multi-Unit Winner Determination Problem with Reserve Prices in Double Auction

Multi-Set of goods G

- n_j copies of good $j \in G$
- market price p_j for good $j \in G$

Set of trades T, with $t = (a, b) \in T$ a bid-ask pair

- a is a utility-quantity pair (u^a, q^a), with $u^a \in \mathbb{R}^+$
- b is a utility-quantity pair (u^b, q^b), with $u^b \in \mathbb{R}^+$
- q^c is a vector of quantities $(q_j^c)_{j \in G}$, for $c \in \{a, b\}$

Allocation $A \subseteq T$
Feasible Allocation $F \subseteq T$

$$\forall j, \sum_{(a,b) \in F} q_j^b \leq n_j + q_j^a$$

Winner Determination = Revenue Maximization

$$\max_{F \subseteq T} \sum_{(a,b) \in F} (u^b - \vec{p} \cdot \vec{q^b}) + (\vec{p} \cdot \vec{q^a} - u^a)$$

Search Algorithms

Minimization in Trees

Blind Search

$\boxed{\text{Breadth-First Search}}$

Heuristic Search

$\boxed{\text{Best-First (Greedy) Search}}$

Optimal Heuristic Search

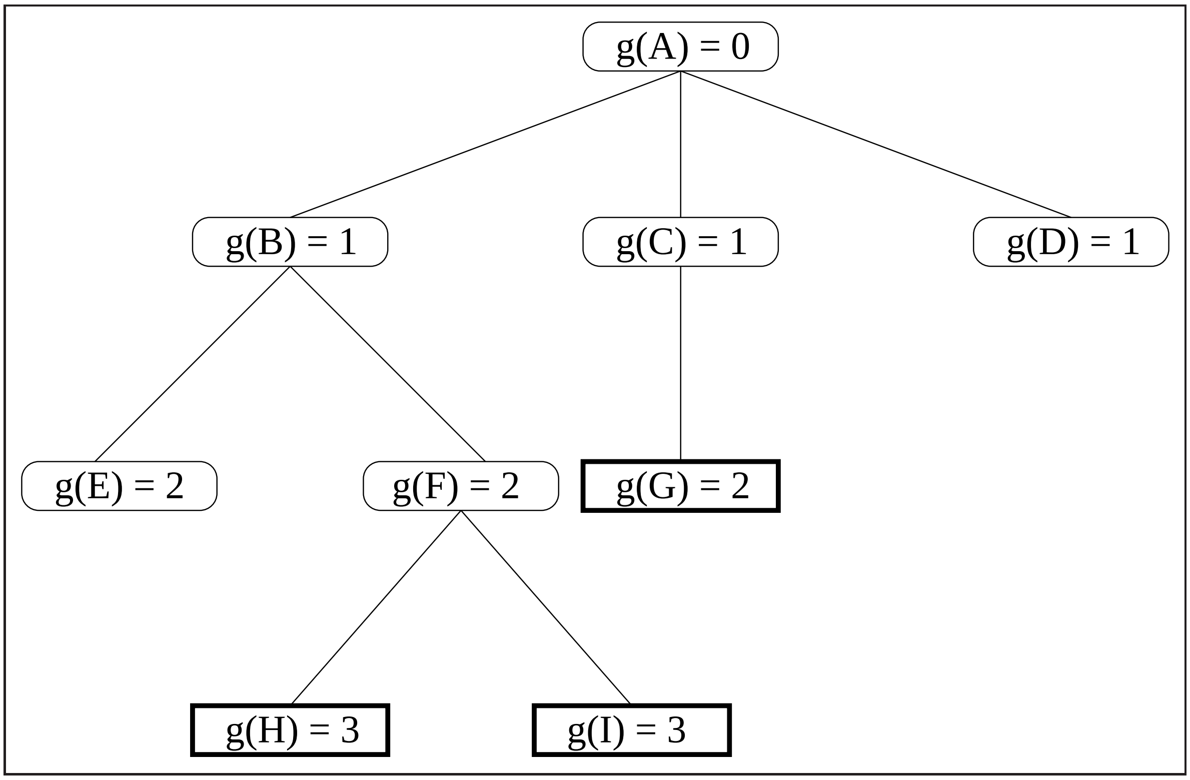
$\boxed{A^* \text{ Search}}$

Approximately Optimal Algorithm

$\boxed{\text{Beam Search}}$

Breadth-First Search

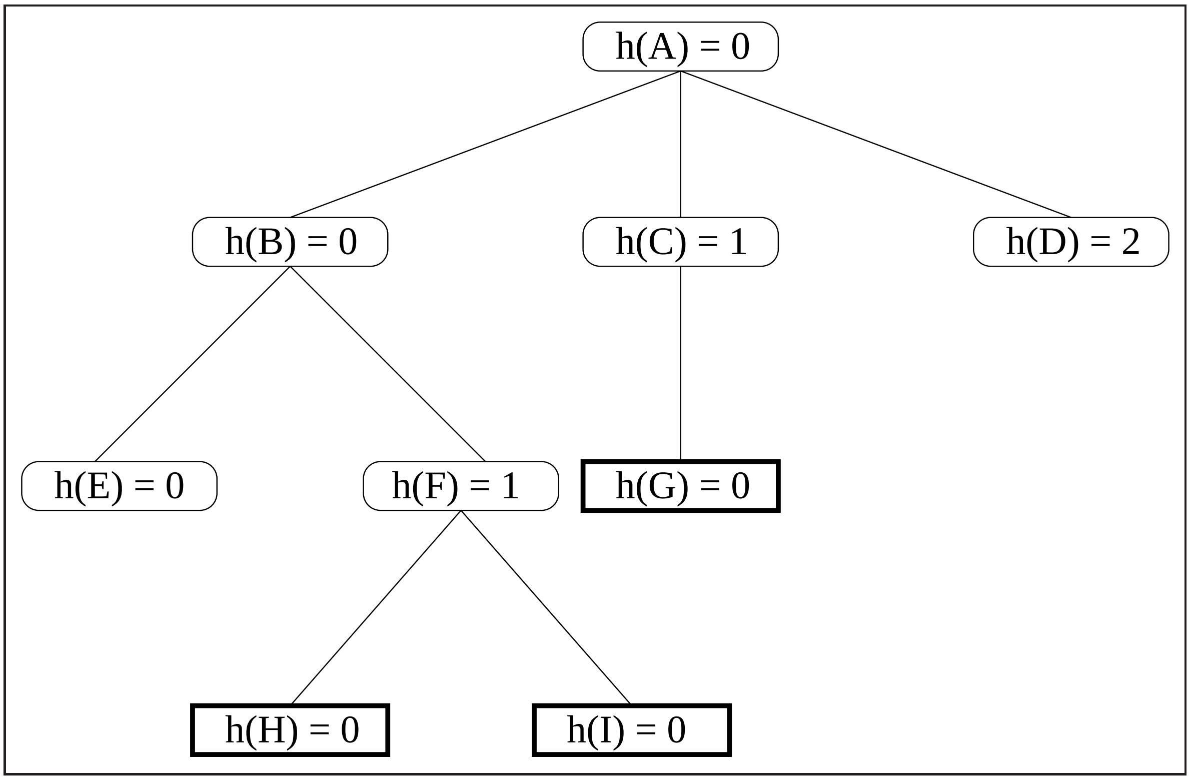
$\Rightarrow$ expand lowest-cost node according to g



Optimal Goal G

Best-First (Greedy) Search

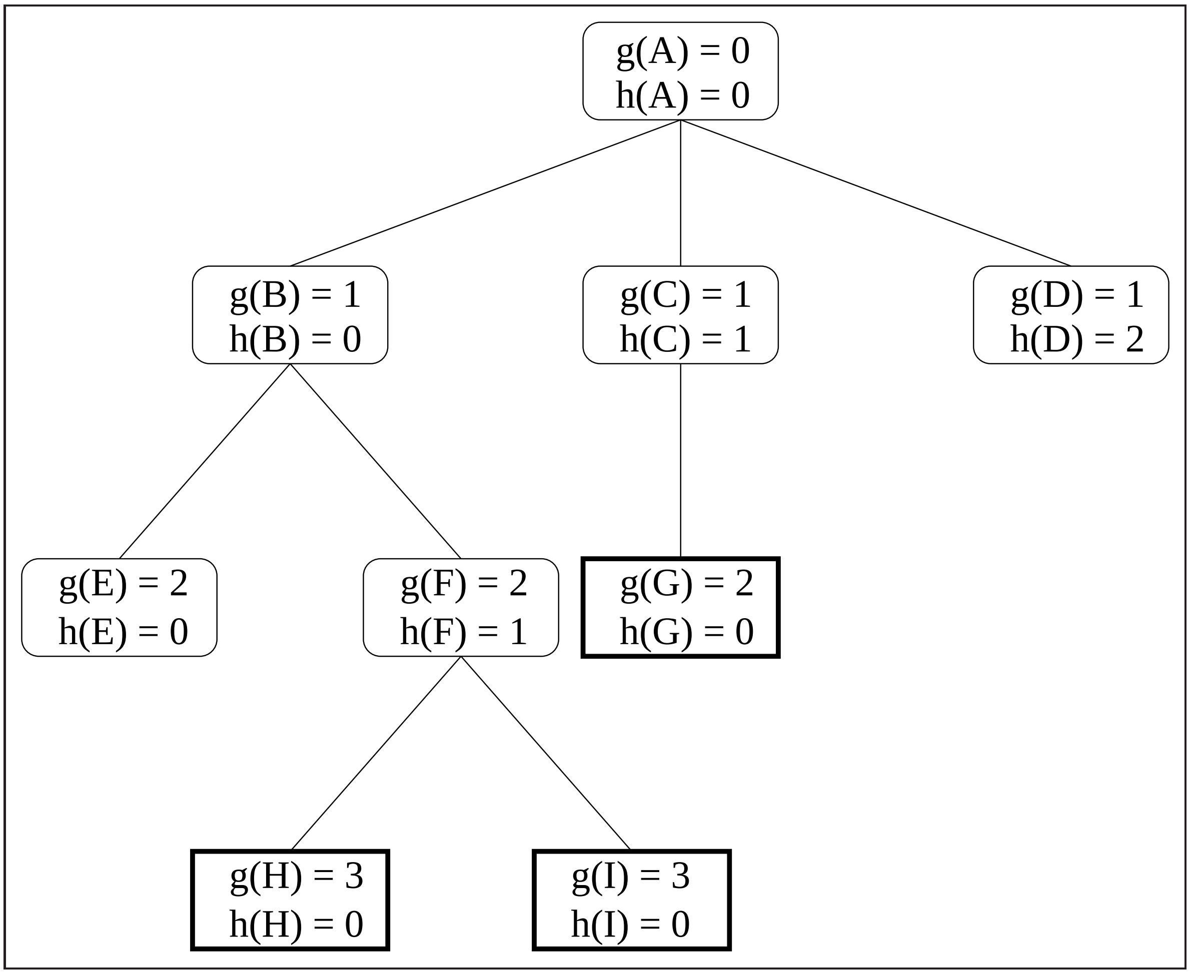
$\Rightarrow$ expand lowest-cost node according to h



Sub-Optimal Goals H, I

A^* Search

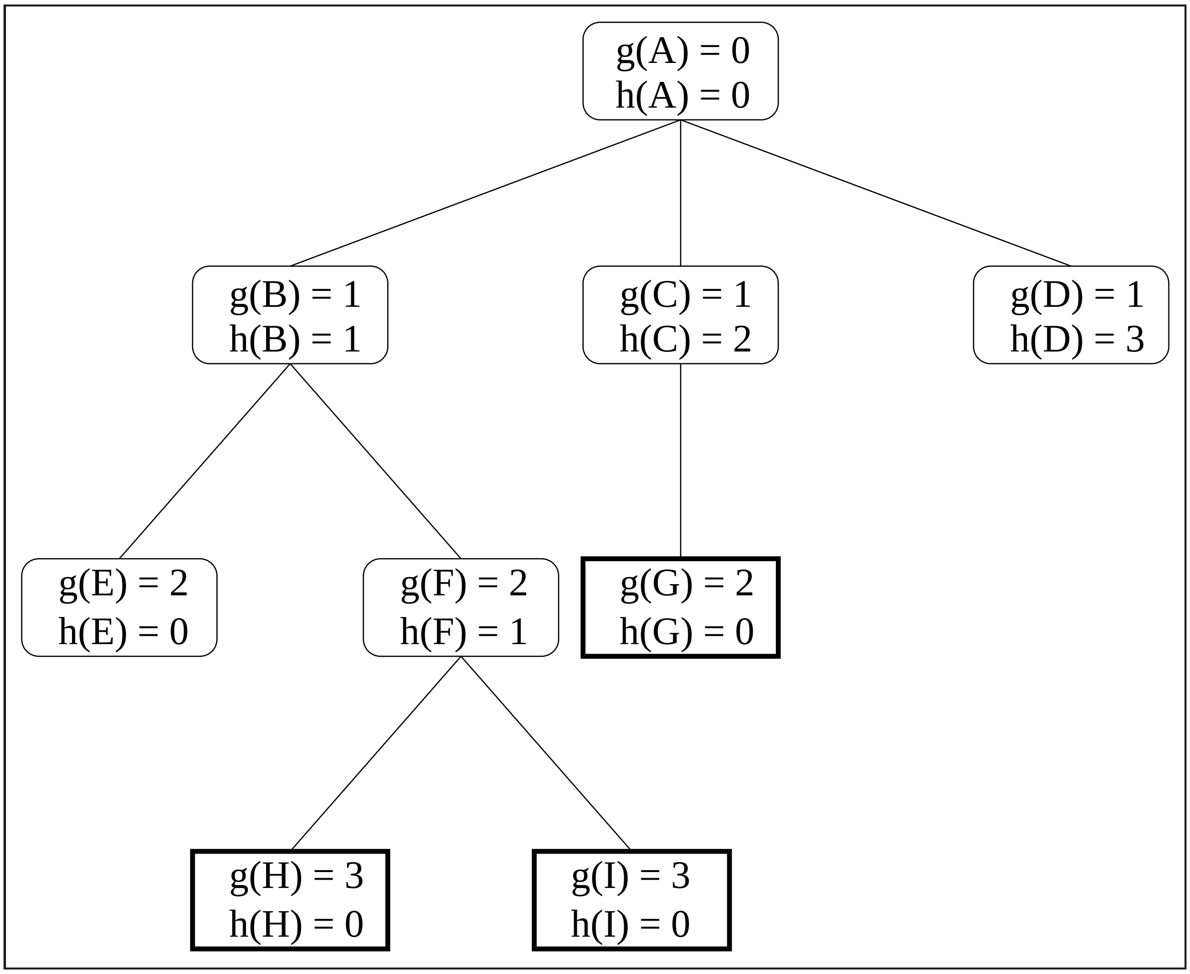
$\Rightarrow$ expand lowest-cost node using $f = g + h$



Optimal Goal G

A^* Search

$\Rightarrow$ expand lowest-cost node using $f = g + h$



Sub-Optimal Goals H, I

Admissible Heuristics

Definition

$\boxed{h \text{ is admissible iff it is optimistic}}$

- minimization $\rightarrow$ underestimate distance to goal
- maximization $\rightarrow$ overestimate value of goal

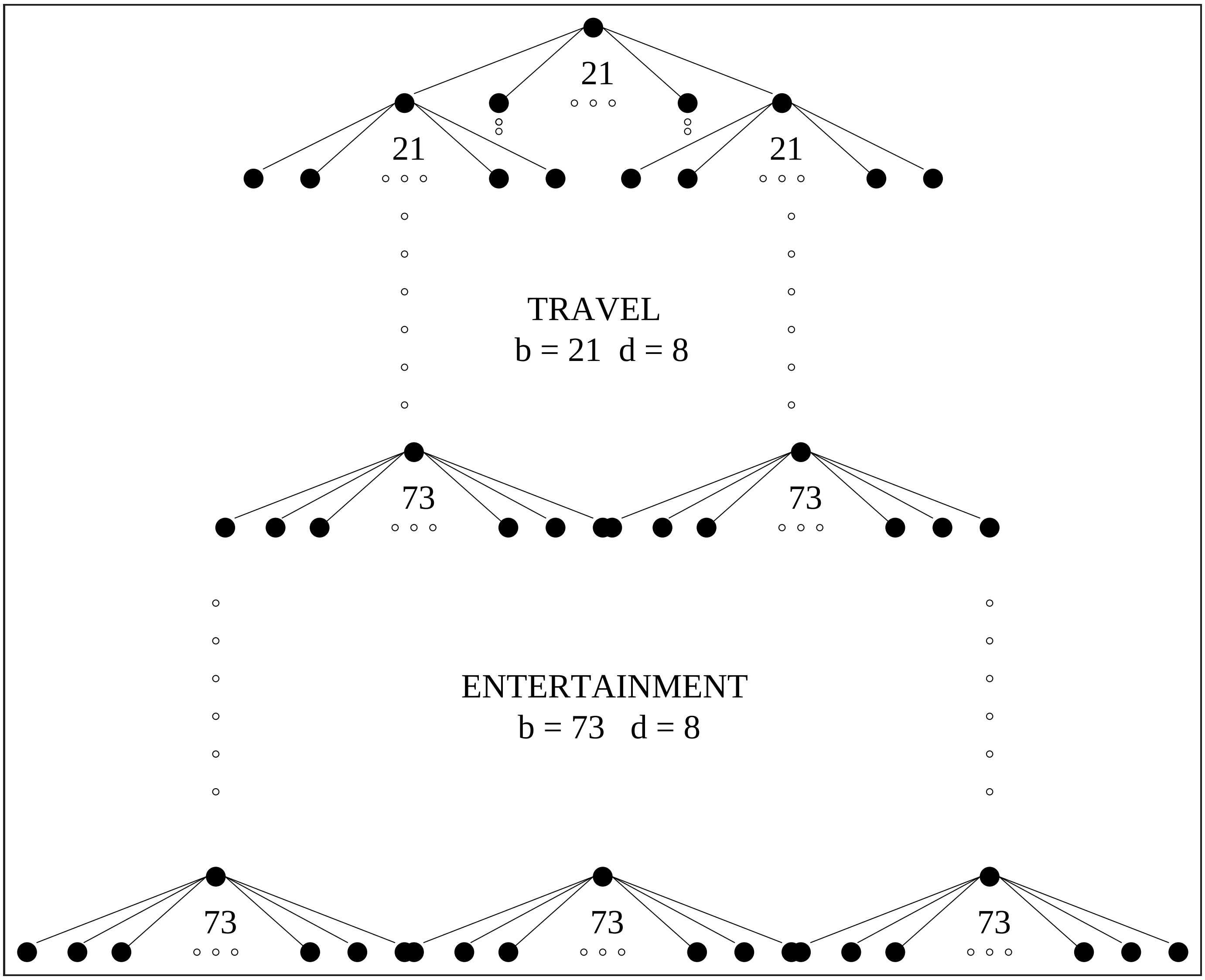
Theorem

$\boxed{A^* \text{ is optimal iff } h \text{ is admissible}}$

Intuition

- pessimism might rule out optimal goal nodes
- optimism never rules out optimal goal nodes

A^* Search Tree



A^* Travel Heuristics

Constraints

CANNOT ASSIGN

1. single client multiple packages
2. single resource to multiple clients
   - single hotel room to multiple clients
   - single arriving flight to multiple clients
   - single departing flight to multiple clients
3. more packages than goods comprise
4. more packages than clients
5. packages w/o goods
6. infeasible packages

Preprocessing

k: upper bound on num packages

c: num clients not assigned travel packages

$k_0 + k_1$: upper bound on num legal hotel packages, double-counting flights

- k_0: upper bound on num packages using flights and bad hotels
- k_1: upper bound on num packages using flights and good hotels

k_2: upper bound on num illegal hotel packages, using flights and both bad and good hotels

$$k = \min(c, k_0 + k_1, k_2)$$

Greedy Counting Algorithm

$\Rightarrow$ compute k_i by greedily counting packages in order from shortest to longest

	in	hot	out
1	1	1	0
2	1	2	2
3	2	2	1
4	0	1	1

2 pkgs of len 1

$\downarrow$

	in	hot	out
1	1	1	0
2	0	1	1
3	1	1	0
4	0	1	1

2 pkgs of len 2

$\downarrow$

	in	hot	out
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0

upper bound $= 4$

Pruning Algorithm

REPEAT

for $d \in \{1, \ldots, 4\}$

1–2. cannot accomodate more arriving/departing clients on day d than num hotels on day d

3. cannot assign more hotel rooms on day d than yesterday's hotels + today's arriving flights

4. cannot assign more hotel rooms on day d than tomorrow's hotels + tomorrow's departing flights

UNTIL quiescence

Pruning Example

	in	hot	out
1	0	2	1
2	3	1	2
3	2	2	3
4	1	1	0

$\xrightarrow{1-2}$

	in	hot	out
1	0	2	1
2	1	1	1
3	2	2	2
4	1	1	0

$\downarrow$3-4

	in	hot	out
1	0	0	0
2	1	1	1
3	2	2	2
4	0	0	0

$\xleftarrow{1-2}$

	in	hot	out
1	0	0	1
2	1	1	1
3	2	2	2
4	1	0	0

Naive Travel Heuristic

$\boxed{\text{Sum top } k \text{ entries}}$

	1	2	3	4	5	6	7	8
12G	850	875	.	.	.	.	.	.
13G	950	975	.	.	.	.	.	.
14G	1050	875	.	.	.	.	.	.
15G	950	775	.	.	.	.	.	.
23G	1050	1075	.	.	.	.	.	.
24G	1150	975	.	.	.	.	.	.
25G	1050	875	.	.	.	.	.	.
34G	1050	875	.	.	.	.	.	.
35G	950	775	.	.	.	.	.	.
45G	850	675	.	.	.	.	.	.
12B	700	800	.	.	.	.	.	.
13B	800	900	.	.	.	.	.	.
14B	900	800	.	.	.	.	.	.
15B	800	700	.	.	.	.	.	.
23B	900	1000	.	.	.	.	.	.
24B	1000	900	.	.	.	.	.	.
25B	900	800	.	.	.	.	.	.
34B	900	800	.	.	.	.	.	.
35B	800	700	.	.	.	.	.	.
45B	700	600	.	.	.	.	.	.

$$\boxed{h(n) = 1925}$$

A^* Travel Heuristics

Insert top `hot[good]` into PQ

	1	2	3	4	5	6	7	8
12G	850	875	.	.	.	.	.	.
13G	950	975	.	.	.	.	.	.
14G	1050	875	.	.	.	.	.	.
15G	950	775	.	.	.	.	.	.
23G	1050	1075	.	.	.	.	.	.
24G	1150	975	.	.	.	.	.	.
25G	1050	875	.	.	.	.	.	.
34G	1050	875	.	.	.	.	.	.
35G	950	775	.	.	.	.	.	.
45G	850	675	.	.	.	.	.	.

Insert top `hot[bad]` into PQ

	1	2	3	4	5	6	7	8
12B	700	800	.	.	.	.	.	.
13B	800	900	.	.	.	.	.	.
14B	900	800	.	.	.	.	.	.
15B	800	700	.	.	.	.	.	.
23B	900	1000	.	.	.	.	.	.
24B	1000	900	.	.	.	.	.	.
25B	900	800	.	.	.	.	.	.
34B	900	800	.	.	.	.	.	.
35B	800	700	.	.	.	.	.	.
45B	700	600	.	.	.	.	.	.

$\boxed{h(n) = 1875}$

A^* Travel Heuristics

Insert top out[2] entries into PQ

	1	2	3	4	5	6	7	8
12G	850	875	.	.	.	.	.	.
12B	700	800	.	.	.	.	.	.

Insert top out[3] entries into PQ

	1	2	3	4	5	6	7	8
13G	950	975	.	.	.	.	.	.
23G	1050	1075	.	.	.	.	.	.
13B	800	900	.	.	.	.	.	.
23B	900	1000	.	.	.	.	.	.

Insert top out[4] entries into PQ

	1	2	3	4	5	6	7	8
14G	1050	875	.	.	.	.	.	.
24G	1150	975	.	.	.	.	.	.
34G	1050	875	.	.	.	.	.	.
14B	900	800	.	.	.	.	.	.
24B	1000	900	.	.	.	.	.	.
34B	900	800	.	.	.	.	.	.

$\boxed{h(n) = 1875}$

A^* Travel Heuristics

Insert top `in`[1] entries into PQ

	1	2	3	4	5	6	7	8
12G	850	875	.	.	.	.	.	.
13G	950	975	.	.	.	.	.	.
14G	1050	875	.	.	.	.	.	.
15G	950	775	.	.	.	.	.	.
12B	700	800	.	.	.	.	.	.
13B	800	900	.	.	.	.	.	.
14B	900	800	.	.	.	.	.	.
15B	800	700	.	.	.	.	.	.

$h(n) = 975$

A^* Travel Heuristics

Partitions

- arrival day 1, 2, 3, 4
- departure day 2, 3, 4, 5
- good hotel, bad hotel

MAIN TRAVEL HEURISTIC

I/P: Partitions $\mathcal{P}$, Bound k

O/P: Estimate $h(n)$

for partition $p \in \mathcal{P}$

  for class $i \in p$

    insert top $\texttt{item}[i]$ entries into PQ

  $\texttt{estimate}[p]$ = sum top k entries in PQ

return minimum $\texttt{estimate}[p]$

A^* Travel Heuristics

- if $\left(\sum_{j \neq i} \mathtt{in}[j] < k\right)$
  then sum top $(k - \mathtt{in}[i])$ class i packages

- if $\left(\sum_{j \neq i} \mathtt{out}[j] < k\right)$
  then sum top $(k - \mathtt{out}[i])$ class i packages

- if $(\mathtt{hot}[\mathtt{good}] < k)$
  then sum top $(k - \mathtt{hot}[\mathtt{good}])$ bad packages

- if $(\mathtt{hot}[\mathtt{bad}] < k)$
  then sum top $(k - \mathtt{hot}[\mathtt{bad}])$ good packages

A^* Entertainment Heuristics

Constraints

CANNOT ASSIGN

1. single client multiple tickets to the same event
2. single client multiple tickets on the same night
3. single ticket to multiple clients
4. entertainment packages w/o tickets
5. entertainment packages inconsistent w/ travel

A^* Entertainment Heuristics

MAIN ENTERTAINMENT HEURISTIC

call entertainment heuristic #1 $\rightarrow$ relax constraint #1
call entertainment heuristic #2 $\rightarrow$ relax constraint #2
call entertainment heuristic #3 $\rightarrow$ relax constraint #3
return minimum estimate

Notation

$\mathtt{ntix}[d,e]$ = number of tickets on date d for event e
$\mathtt{ntix}[d] = \sum_e \mathtt{ntix}[d,e]$ = number of tickets on date d
$\mathtt{ntix}[e] = \sum_d \mathtt{ntix}[d,e]$ = number of tickets to event e

Naive Entertainment Heuristic

ASSUME

$\boxed{\text{ASSIGN } \texttt{ntix}\ [d, e] \text{ TO CLIENTS IN TOWN ON DATE } d \text{ WHO MOST VALUE EVENT } e}$

	BRS	SY	PH
1	150	100	50
2	25	50	75

$\texttt{tix} = \{\mathrm{BRS1}, \mathrm{SY1}, \mathrm{BRS2}, \mathrm{SY2}\}$

`intown[1,1]`, `intown[1,2]`

`intown[2,1]`, `intown[2,2]`

$\boxed{h(n) = 500}$

Entertainment Heuristic #1

ASSUME

$\boxed{\text{for all } d\text{, ASSIGN } \texttt{ntix}\ [d] \text{ TO CLIENTS IN TOWN ON DATE } d \text{ WHO MOST VALUE EVENTS FOR WHICH TICKETS ARE OWNED ON THAT DATE}}$

	BRS	SY	max
1	150	100	150
2	25	50	50

$\texttt{tix} = \{\text{BRS1}, \text{SY1}, \text{BRS2}, \text{SY2}\}$

`intown[1,1]`, `intown[1,2]`
`intown[2,1]`, `intown[2,2]`

$\boxed{h(n) = 400}$

Entertainment Heuristic #2

ASSUME

for all e, ASSIGN `ntix` $[e]$ TO CLIENTS WHO MOST VALUE EVENT e AND WHOSE NIGHTS IN TOWN INTERSECT NIGHTS WITH TICKETS FOR EVENT e

	BRS	SY	PH
1	150	100	50
2	25	50	75

$\mathtt{tix} = \{\mathrm{BRS1}, \mathrm{SY1}, \mathrm{BRS2}, \mathrm{SY2}\}$

`intown[1,1]`, `intown[1,2]`
`intown[2,1]`, `intown[2,2]`

$\boxed{h(n) = 325}$

Entertainment Heuristic #3

ASSUME

$\boxed{\text{ASSIGN EACH CLIENT ITS PREFERRED ENTERTAINMENT PACKAGE THAT IS COMPATIBLE WITH ITS TRAVEL}}$

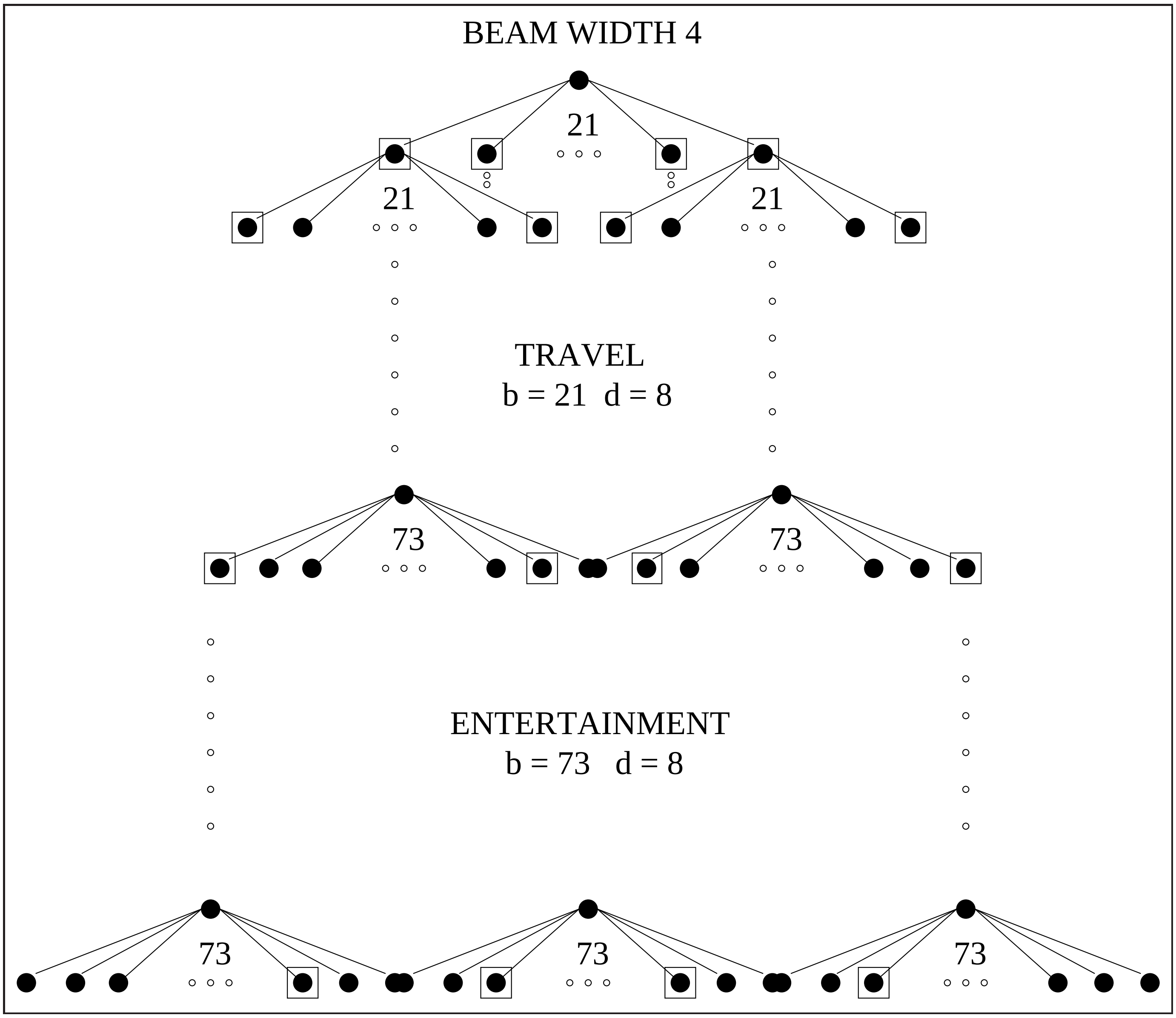
	BRS	SY	PH
1	150	100	50
2	25	50	75

$\mathtt{tix} = \{\mathrm{BRS1}, \mathrm{SY1}, \mathrm{BRS2}, \mathrm{SY2}\}$

`intown[1,1]`, `intown[1,2]`
`intown[2,1]`, `intown[2,2]`

$\boxed{h(n) = 325}$

Beam Search


BEAM WIDTH 4
21
21
21
TRAVEL
b = 21 d = 8
73
73
ENTERTAINMENT
b = 73   d = 8
73
73
73

Priceline Data Structure

S: total supply
D: total demand
H: current holdings

pre-priceline $\vec{q}$: $\langle q_1, \ldots, q_{S+D} \rangle$
priceline $\vec{p}$: shift $\vec{q}$ to the left by $D - H$ entries
shift $\vec{q}$ to the right by $H - D$ entries

- $S = \infty,\ D = H = 0$
  $\vec{q} = \vec{p} = \langle 315, 315, \ldots \rangle$

- $S = \infty,\ D = 0,\ H = 2$
  $\vec{q} = \langle 315, 315, \ldots \rangle$
  $\vec{p} = \langle 0, 0, 315, 315, \ldots \rangle$

- $S = 16,\ D = H = 0$
  $\vec{q} = \vec{p} = \langle 105, 155, 205, 255, 305, 355, 405, \infty, \ldots, \rangle$

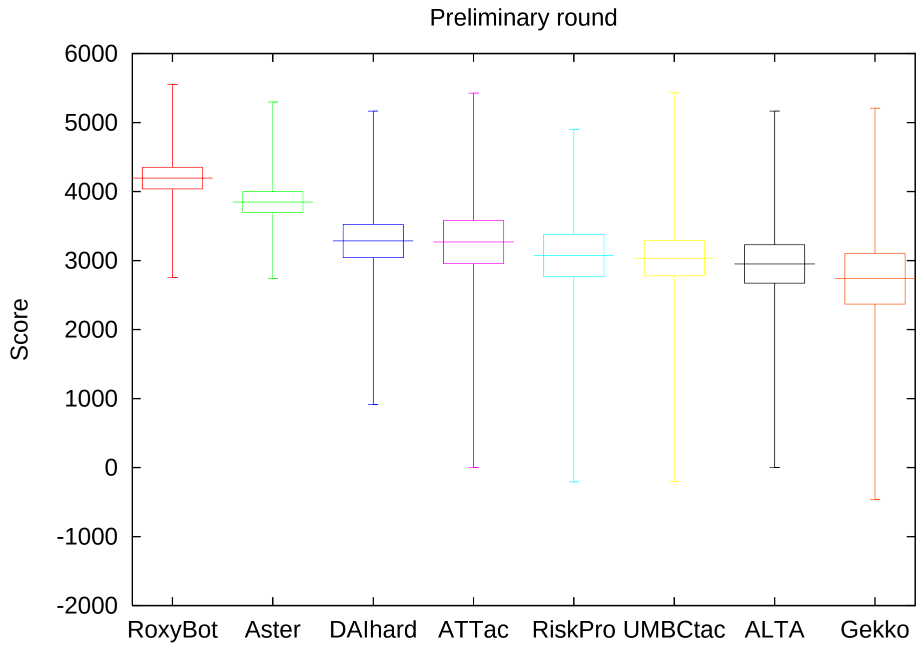
- $S = 2,\ D = 2,\ H = 4$
  $\vec{q} = \langle 25, 65, 75, 115, \infty, \ldots \rangle$
  $\vec{p} = \langle 0, 0, 25, 65, 75, 115, \infty, \ldots \rangle$

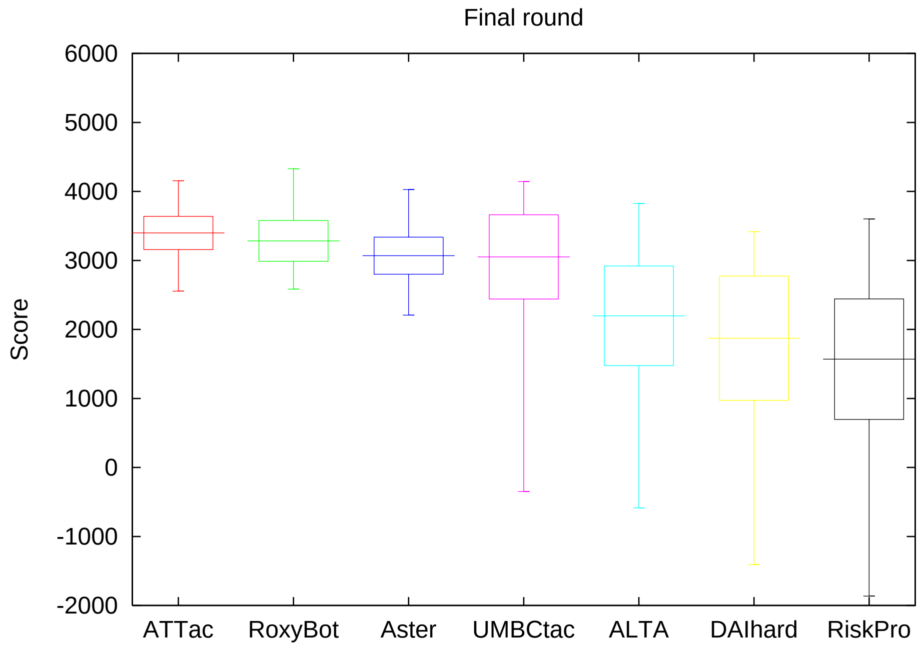
- $S = 2,\ D = 2,\ H = -1$
  $\vec{q} = \langle 25, 65, 75, 115, \infty, \ldots \rangle$
  $\vec{p} = \langle 115, \infty, \ldots \rangle$

Finalists

Agent	Designers
`ATTac`	Peter Stone, Michael Littman, Satinder Singh, Michael Kearns
`RoxyBot`	Justin Boyan, Amy Greenwald
`Aster`	Andrew Goldberg, Umesh Maheshwari
`UMBCTac`	Youyong Zou
`ALTA`	Andrey Tarkhov, Dmitry Uspensky, Eugene Vostroknautov
`DAIHard`	Rajatish Mukherjee, Partha Dutta, Sandip Sen
`RiskPro`	Magnus Boman, Sven-Erik Ceedigh
`T1`	Lars Olsson, Erik Aurell, Lars Rasmusson, Martin Aronsson, Per Larsson, Glenn Lawer

Final Statistics





Conclusions

Heuristic Search

- A^* Search
  - optimal
  - memory-hog
- Beam Search
  - fast execution
  - approximately optimal
- non-linear objective functions
- anytime solutions

Current Experiments

- Speed of A^* Search vs. Integer LP
- Optimality of Beam Search

Next Year, Hopefully

ESTIMATION *a.k.a* Machine Learning