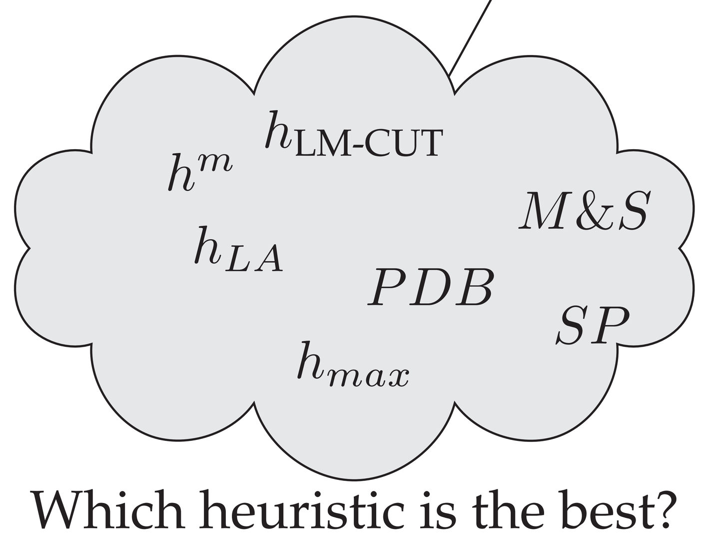
To Max or not to Max: Online Learning for Speeding Up Optimal Planning

Carmel Domshlak, Erez Karpas, Shaul Markovitch - Technion

Motivation

The problem of interest: optimal planning, in a time-bounded setting

$$A^* \equiv f = g + h$$



Which heuristic is the best?

Choosing a Heuristic

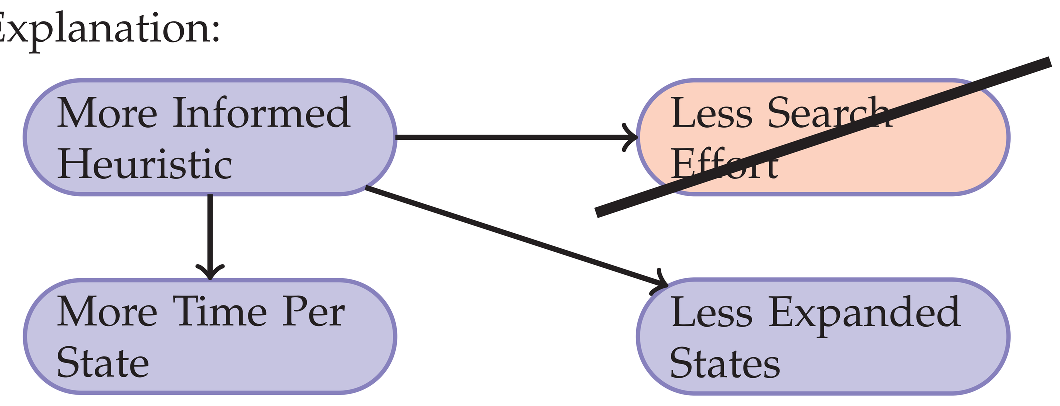
- No single best heuristic ⇒ why only one?
- Use the maximum of several heuristics

Sample results:

Domain	h_{LA}	$h_{\text{LM-CUT}}$	max
airport	25	38	36
freecell	28	15	22

Number of problems solved in 30 minutes

Explanation:



Simple Observation

The maximum of several heuristics — is simply the value of one of those heuristics. Idea:

- Choose $h_i = \text{ORACLE}(s, \{h_1, \ldots, h_n\})$
- Compute only $h_i(s)$

How do we define ORACLE?

$$\text{ORACLE}(s, \{h_1, \ldots, h_n\}) \stackrel{?}{=} \operatorname*{argmax}_i h_i(s)$$

- The extra time to compute the most informed heuristic may not be worth it
- We need to come up with a theoretical model

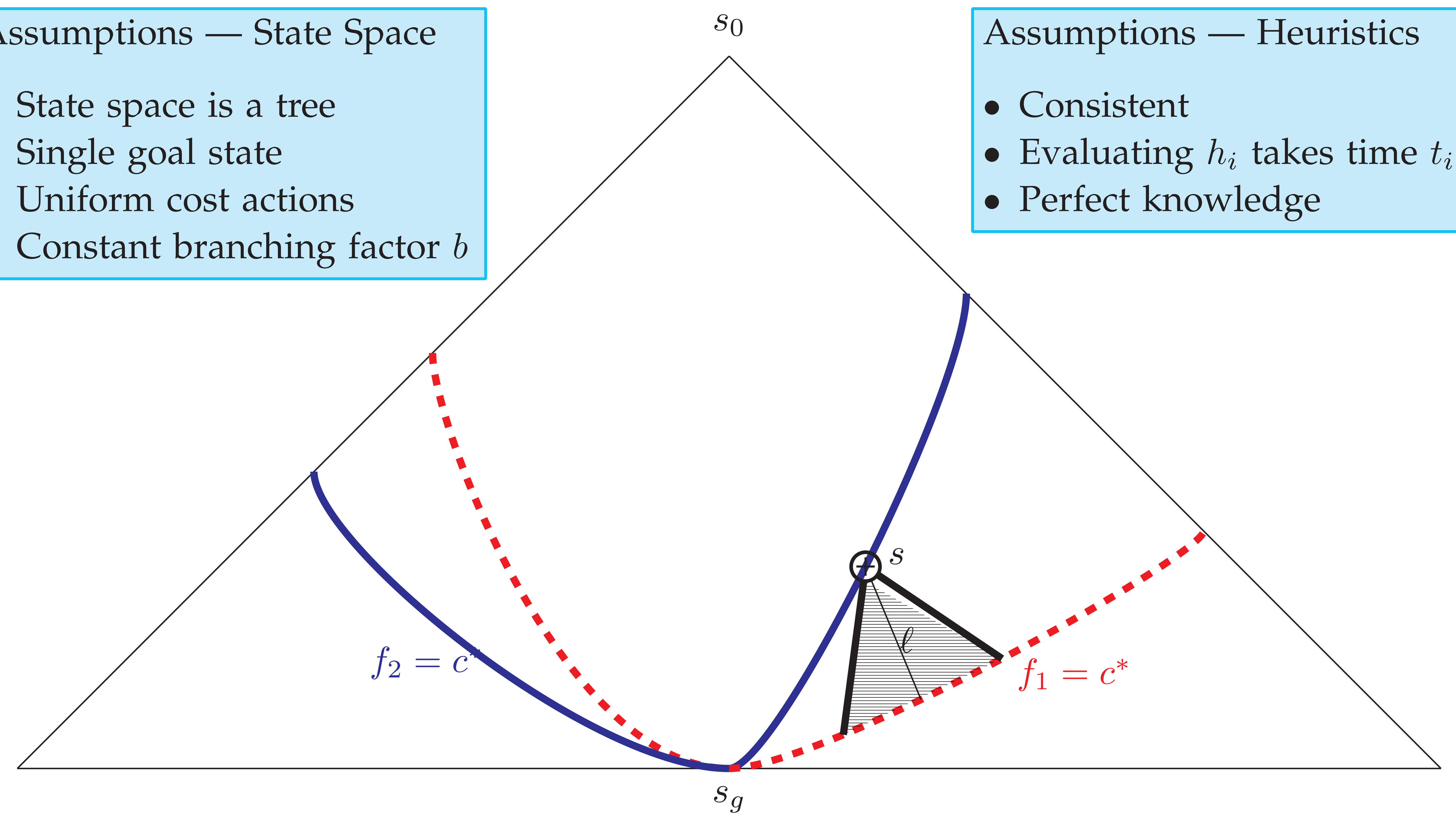
Theoretical Model Illustrated

Assumptions — State Space

- State space is a tree
- Single goal state
- Uniform cost actions
- Constant branching factor b

Assumptions — Heuristics

- Consistent
- Evaluating h_i takes time t_i
- Perfect knowledge



Decision Rule

In the surely expanded region (above both lines) — just expand, don't evaluate.

For state s on the border, either:

- use h_2, which takes time t_2, or
- use h_1, in which case we expand the highlighted region in $b^\ell t_1$ time

Best decision for s: use h_2 iff

$$t_2 < b^\ell t_1 \iff \ell > \log_b(t_2/t_1)$$

After some more assumptions about the rate of growth of heuristic value

Use h_2 iff

$$h_2 - h_1 > \alpha \log_b(t_2/t_1)$$

α is a hyper-parameter

Dealing with Assumptions

Assumptions

- State space is a tree - rule is still applicable
- Single goal state - rule is still applicable
- Uniform cost actions - rule is still applicable
- Constant branching factor b - estimate
- Perfect knowledge - use decision rule at every state

Two heuristics: h_1 and h_2

- Consistent - rule is still applicable
- Evaluating h_i takes time t_i - estimate

Labelling Examples

1. b, t_1, t_2 are estimated
2. $h_2 - h_1$ is calculated for each state
3. A state is labeled by h_2 iff

$$h_2 - h_1 > \alpha \log_b(t_2/t_1).$$

WLOG $t_2 > t_1$ - the choice is always whether to evaluate the more expensive heuristic

Experiments

We used two state of the art heuristics:

- $h_{\text{LM-CUT}}$ (Helmert and Domshlak 2009)
- h_{LA} (Karpas and Domshlak 2009)

Parameters:

- $\alpha = 1$
- $\rho = 0.6$
- Training set size = 100

Problems from 22 domains from IPC 1–5

Learning

Pre-search:

- Collecting training examples
- Labeling training examples
- Generating features
- Building a classifier

During search:

- Classification
- Active learning

Features

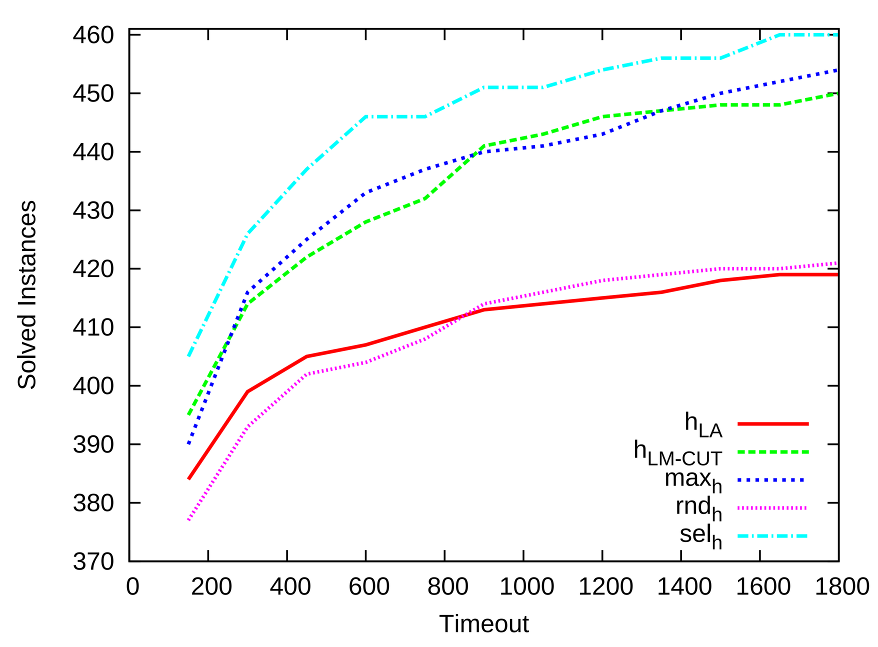
The simplest features — just values of state variables — are used

Classifier

We use the Naive Bayes classifier, because it is:

- Very fast
- Incremental
- Provides probabilistic classification

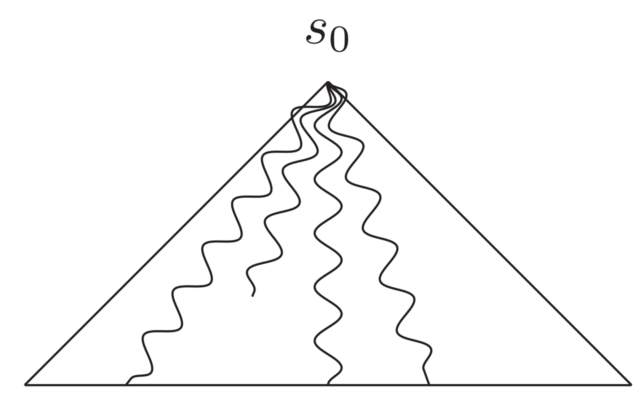
Anytime Behavior



Collecting Examples

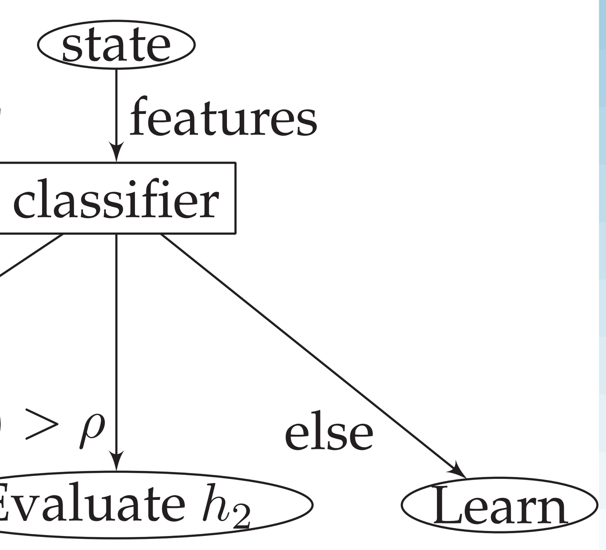
Sample using stochastic hill climbing "probes"

- Depth limit = $2 * h(s_0)$
- Expand s with probability $\sim 1/h(s)$



Using the Learned Model

ρ is a hyper-parameter



Total Time