# A Beginner's Introduction to Heuristic Search Planning

## 4. Five Families of Heuristics

Malte Helmert Gabriele Röger

AAAI 2015 Tutorial

January 25, 2015

# Heuristics

# Planning as Heuristic Search

**Planning as Heuristic Search**

general search algorithm (e.g. $A^*$, greedy best-first search)

+ distance estimator (heuristic function)

# Desirable Properties of Heuristics for Optimal Planning

heuristic $h$ maps states to numbers in $\mathbb{R}_{\geq 0} \cup \{\infty\}$

Desirable properties for optimal planning:

- admissibility: $h(s) \leq h^*(s)$
  for all states $s$
- consistency: $h(s) \leq cost(o) + h(s')$
  for all state transitions $s \xrightarrow{o} s'$

$h^*$: actual distance to goal ("perfect heuristic")

# Desirable Properties for All Heuristics

Desirable property for optimal and satisficing planning:

- accuracy: $h(s)$ should be "close" to $h^*(s)$

# The Challenge

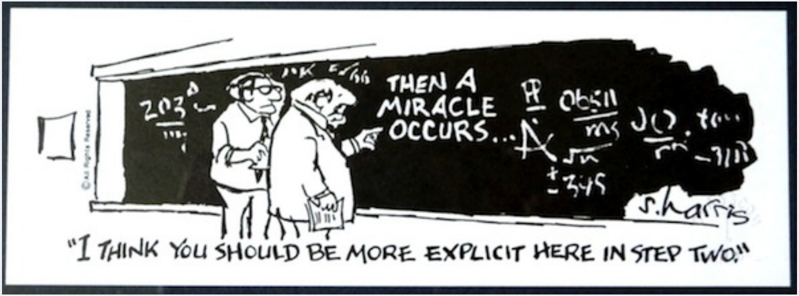

How do we come up with precise estimates in a domain-independent fashion?

# How to Come Up with Good Heuristics

How do we come up with good heuristics?

A commonly held view:

## Inspiration




Must we wait for inspiration to strike?

# How to Come Up with Heuristics

How do we come up with good heuristics?

Another commonly held view:

**Perspiration**

"None of my inventions came by accident. I see a worthwhile need to be met and I make trial after trial until it comes. What it boils down to is one per cent inspiration and ninety-nine per cent perspiration."

— Thomas Alva Edison (1929)

Phrased less positively:
"Throw enough mud at the wall, some of it will stick."

# How to Come Up with Heuristics

How do we come up with good heuristics?

Our recommendation:

**Careful Scientific Study**

"First, you have to understand the problem."

— George Pólya (1945)

# The Science of Heuristics

Ask (and answer) questions:

- Dissect existing approaches: where do they work? Where not? Why and why not?
- Can specific approaches be distilled to general ideas? Can general ideas be applied more specifically?
- Compare existing approaches: does one dominate another? What do they have in common? How are they different? Can their strengths be combined?

# Five Families of Heuristics

# Five Families of Heuristics

How do we come up with heuristics for general problems?

$\rightsquigarrow$ five major approaches in the literature

- delete relaxation
- abstraction
- critical paths
- landmarks
- network flows

# Running Example: FreeCell

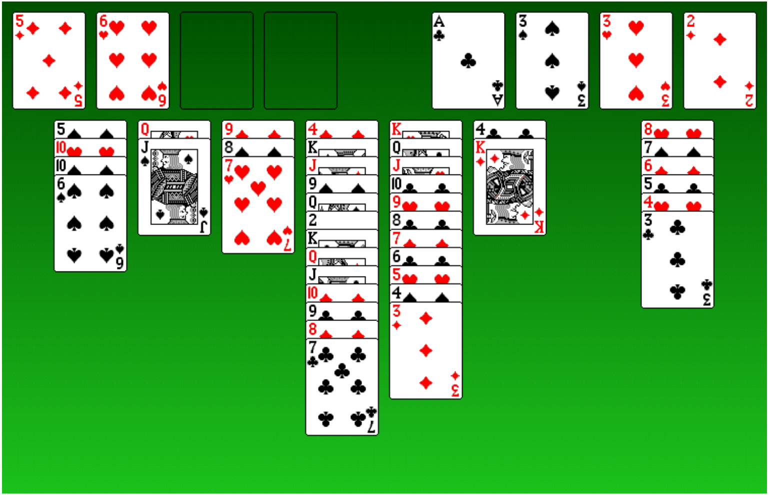


image credits: GNOME Project (GNU General Public License)

# Heuristics in Fast Downward

When a heuristic is implemented in Fast Downward,
we mention its plug-in name.

- Many heuristics have options, sometimes with bad defaults.
- We use dots (e.g., `lmcount(...)`) when options matter a lot.

Want to find out more?

- `http://www.fast-downward.org/Doc/Heuristic`
- Check out the cited papers.
- public mailing list
  $\leadsto$ link on `http://www.fast-downward.org`
- Ask us!

# Planning Heuristics: Delete Relaxation

Five classes of heuristics:

**1. Delete Relaxation**

Estimate cost to goal by considering simpler planning task without negative side effects of actions.

**Example: Delete Relaxation in FreeCell**

Problem constraints dropped by the delete relaxation in FreeCell:

- free cells and free tableau positions remain available after moving cards into them
- cards remain movable and remain valid targets for other cards after moving cards on top of them

⇝ more in Part 6

# Delete Relaxation Heuristics in the Literature

Delete relaxation heuristics in the literature (admissible in red):

- max heuristic (Bonet & Geffner, ECP 1999; AIJ 2001)
- additive heuristic (Bonet & Geffner, ECP 1999; AIJ 2001)
- FF heuristic, $h^+$ heuristic (Hoffmann & Nebel, JAIR 2001)
- pairwise max heuristic (Mirkis & Domshlak, ICAPS 2007)
- set-additive heuristic (Keyder & Geffner, ECAI 2008)
- Steiner tree heuristic (Keyder & Geffner, IJCAI 2009)
- landmark-cut heuristic (Helmert & Domshlak, ICAPS 2009)
- improved LM-Cut (Bonet & Helmert, ECAI 2010)

**in Fast Downward**

`hmax()`, `add()`, `ff()`, `lmcut()`

# Planning Heuristics: Abstraction

Five classes of heuristics:

**2. Abstraction**

Estimate cost by projecting the state space to a smaller space (applying a graph homomorphism).

**Example: Abstraction in FreeCell**

One possible abstraction for FreeCell:
project away all cards that are not 10s, Js, Qs or Ks.

$\rightsquigarrow$ more in Part 5

# Abstraction Heuristics in the Literature

Abstraction heuristics in the literature (admissible in red):

- pattern databases (PDBs) (Edelkamp, ECP 2001; Haslum et al., AAAI 2007; Pommerening et al., IJCAI 2013)
- symbolic PDBs (Edelkamp, AIPS 2002)
- constrained PDBs (Haslum et al., AAAI 2005)
- merge-and-shrink (Dräger et al., SPIN 2006; Helmert et al., ICAPS 2007; Sievers et al. AAAI, 2014)
- structural patterns (Katz & Domshlak, ICAPS 2008)
- Cartesian abstraction (Seipp & Helmert, ICAPS 2013; ICAPS 2014)

**in Fast Downward**

`pdb(...)`, `zopdbs(...)`, `cpdbs(...)`, `gapdb(...)`, `ipdb(...)`, `merge_and_shrink(...)`

# Planning Heuristics: Critical Paths

Five classes of heuristics:

## 3. Critical Paths

Estimate cost by critical path length (makespan) of a concurrent solution for a simplified problem.

In the simplification, a set of subgoals is considered reachable when all size-$m$ subsets are reachable; $m \in \mathbb{N}_1$ is a parameter.

## Example: Critical Paths in FreeCell

Possible critical path for single subgoals ($h^1$):

- Solving the FreeCell task requires four subgoals: have each of $\diamondsuit K$, $\heartsuit K$, $\spadesuit K$, $\clubsuit K$ at foundations
- Follow third subgoal: getting $\spadesuit K$ to foundations requires first having $\spadesuit Q$ at foundations and having $\spadesuit K$ movable.
- Follow second subsubgoal: having $\spadesuit K$ movable requires. . .

# Critical Path Heuristics in the Literature

Critical path heuristics in the literature (admissible in red):

- *$h^m$* heuristic family (Haslum & Geffner, AIPS 2000)
- additive $h^m$ (Haslum et al., AAAI 2005)
- additive-disjunctive heuristic graphs (Coles et al., ICAPS 2008)

**in Fast Downward**

`hm(m=2)`, `hm(m=3)`, ...
warning: very inefficient implementation!

# Planning Heuristics: Landmarks

Five classes of heuristics:

**4. Landmarks**

An action set $A$ is a landmark if all plans include an action from $A$.

Compute a set of landmarks and use it to derive a cost estimate (e.g., by counting the number of landmarks).

**Example: Landmarks in FreeCell**

Landmarks in FreeCell:

- The set of actions that move the $\heartsuit Q$ to foundations.
- The set of actions that move the $\clubsuit 7$ away from the $\diamondsuit 8$.
- ...

$\rightsquigarrow$ more in Part 6

# Landmark Heuristics in the Literature

Landmark heuristics in the literature (admissible in red):

- LAMA heuristic (Richter et al., AAAI 2008)
- cost-partitioned landmarks (Karpas & Domshlak, IJCAI 2009)
- conjunctive landmarks (Keyder et al., ECAI 2010)
- landmark-cut heuristic (Helmert & Domshlak, ICAPS 2009)

**in Fast Downward**

`lmcount(...)`, `lmcut()`

# Planning Heuristics: Network Flows

Five classes of heuristics:

## 5. Network Flows

In every plan, the number of times each fact is produced
vs. the number of times it is consumed must be "balanced".
Solve a linear program encoding this information.

## Example: Flow Constraints in FreeCell

Flow constraints in FreeCell:

- Each card is moved into a free cell
  as often as it is moved out of a free cell.
- Each card is buried in a stack as often as it is uncovered.
- ...

(Give or take a bit to account for current state and goal.)

# Network Flow Heuristics in the Literature

Network flow heuristics in the literature (admissible in red):

- flow heuristic (van den Briel et al., CP 2007)
- state equation heuristic (Bonet, IJCAI 2013)
- enhanced flow heuristics (Bonet & van den Briel, ICAPS 2014)

# Summary

# Summary

- the challenge: precise domain-independent heuristics
- it bears repeating: heuristics can be studied scientifically
- five major approaches in the literature:
    - delete relaxation
    - abstraction
    - critical paths
    - landmarks
    - network flows