Advanced Results on Distance Estimation in Planning

1. Overview of the Area and Its Recent Results

What We Will, and Won't, Talk About

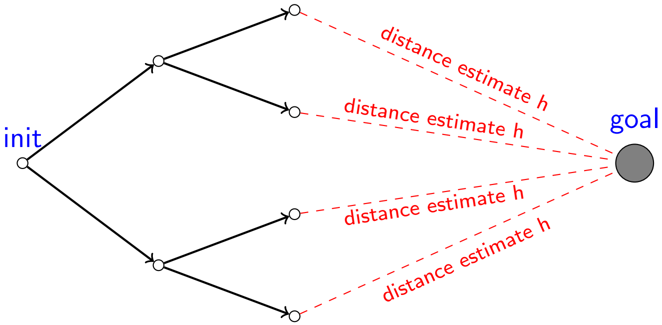
Jörg Hoffmann and Michael Katz

SAARLAND UNIVERSITY
COMPUTER SCIENCE

Summer 2013

Agenda

What We Consider: Heuristic (Forward) Search

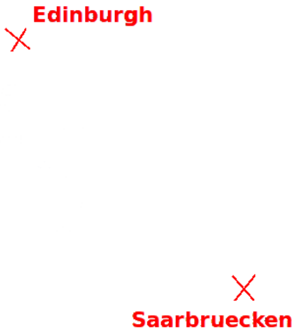


→ Heuristic function h maps world states s to an estimate $h(s)$ of goal distance. Search prefers to explore states with small h.

What We Consider: Heuristic Functions

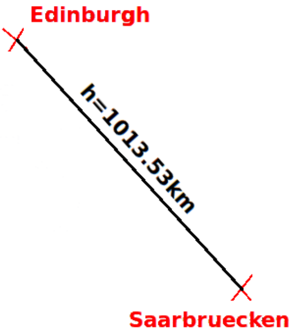
Problem: Find a route from Saarbruecken To Edinburgh.

What We Consider: Heuristic Functions



Simplified Problem: Throw away the map.

What We Consider: Heuristic Functions



Heuristic function: Straight line distance.

What We Consider: Heuristic Functions

$\rightarrow$ Heuristic functions h are computed as solutions to simplified versions of the problem at hand.

How to do this in planning?

- The "problem at hand" is anything that can be described in the declarative input language.
- We will consider STRIPS and Finite-Domain Representation (FDR) (aka "SAS+", multi-valued variables) interchangeably.
- We want to generate h fully automatically, given only that input.

  $\rightarrow$ It's a long way to the goal, but how long exactly?

$\rightarrow$ For simplicity, we will mostly stick to distance, i.e., plan length. Most of what we'll talk about can be done for arbitrary additive action costs.

Why Consider Heuristic Search Planning?

IPC = The International Planning Competition:

- **IPC 2000 Winner:** heuristic search.
- **IPC 2002 Winner:** heuristic search.
- **IPC 2004 Winner:** satisficing: heuristic search, optimal: SAT.
- **IPC 2006 Winner:** satisficing: heuristic search, optimal: SAT.
- **IPC 2008 Winner:** satisficing: heuristic search, optimal: symbolic search.
- **IPC 2011 Winner:** satisficing: heuristic search (first 12 places), optimal: heuristic search (first 9 places).

ATTENTION!

- This is only for the fully-automatic deterministic tracks of the IPC.
- This does NOT mean heuristic search is universally better; it's only the IPC setup.
- "Winner" is a very inadequate summary of such huge and complex events.

→ All I'm saying is: This approach has been mainstream in academic planning research during the last decade, and has produced a lot of interesting results.

Warning!!

This is an advanced lecture. If you're not at least vaguely familiar with heuristic search planning, you're not likely to understand much.

- In this introductory lecture, I will give a brief overview of the area, that you should be able to follow in any case. (Although you won't be able to answer my questions if you're a beginner.)
- In the technical lectures, we will assume familiarity with the basics.
- Sorry, but if we cover all the basics in detail, then we won't get around to say much about the recent stuff.

The 4 Families (That We Know Up To Now)

Ignoring Deletes

$h^{\max}$
h^{+}
h^{add}

Abstractions

PDB
M&S

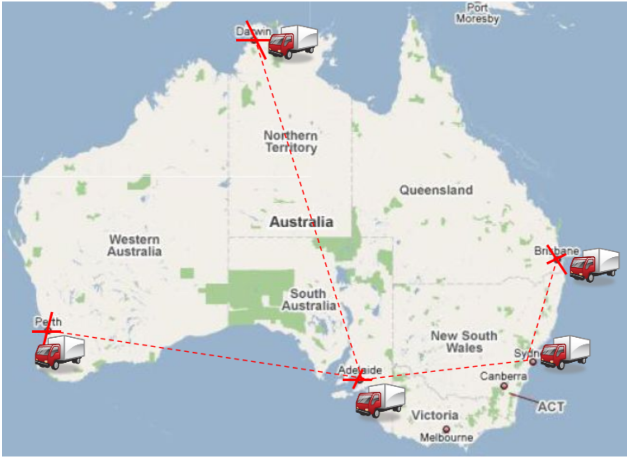
Critical Paths

h^1
h^2
h^3
...

Landmarks

h_L^{LM}
(Elementary LM h)

Ignoring Deletes: Example



$\rightarrow h =$ **Minimum Spanning Tree**

Ignoring Deletes: Details

- How is h^+ defined? (in STRIPS)

  $\rightarrow$ Given a world state s, $h^+(s)$ is the length of an optimal relaxed plan for s, i.e., a plan for s in the relaxed task where the delete lists are assumed to be empty.

- Can we compute h^+ efficiently?

  $\rightarrow$ No, the corresponding decision problem is **NP**-hard.

- If not, what approximations are known?

  $\rightarrow$ h^{max} approximates the cost of a set of > 1 facts by the cost of the most difficult single fact in the set; h^{add} instead approximates the cost of the set by the sum of the costs of its facts. The relaxed plan heuristic generates some not necessarily optimal plan for the relaxed task.

- How is h^+ defined for FDR planning?

  $\rightarrow$ In FDR, ignoring deletes means assuming that state variables accumulate their values, rather than switching between them.

Ignoring Deletes: (Some) Recent Results

"Recent": The last 5 or so years.

"Some": A sample of results that we personally find important.

(a) **Automatic h^+ search space surface analysis [Hoffmann (2011)]:** One can identify classes of planning tasks whose surface has particular properties (absence of local minima) based on properties of the causal graph and the domain transition graphs. This connection can be exploited for automatic analysis predicting "how difficult" a task is for delete relaxation heuristics.

(b) **Marriage of h^m with h^+ [Haslum (2012); Keyder et al. (2012)]:** Allows to interpolate between h^+ and h^*; see below under critical-path heuristics.

(c) **Relaxing only some of the state variables [Katz et al. (2013b,a)]:** Red variables accumulate their values, black variables switch between them. Allows to interpolate between h^+ and h^*.

→ All of (a–c) are covered in Lecture 2.

The 4 Families (That We Know Up To Now)

Ignoring Deletes

h^{max}
h^{+}
h^{add}

Abstractions

PDB
M&S

Critical Paths

h^{1}
h^{2}
h^{3}
...

Landmarks

h_{L}^{LM}
(Elementary LM h)

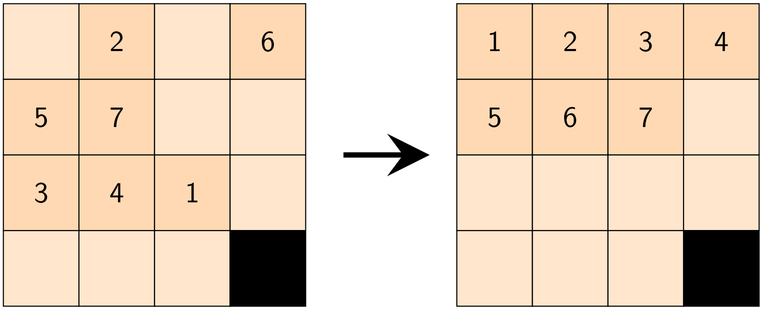
Abstractions: Example

9	2	12	6
5	7	14	13
3	4	1	11
15	10	8	

→

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Abstractions: Example



$\rightarrow h =$ **Solution to Smaller (and Easier) Puzzle**

Abstractions: Details

- What is an abstraction, formally?

  $\rightarrow$ An abstraction is a function α mapping between the state space (all world states) and a (smaller) set of abstract world states.

- How is the corresponding heuristic function h^α defined?

  $\rightarrow$ Given a world state s, $h^\alpha(s) = h^*_{\theta^\alpha}(\alpha(s))$ where $h^*_{\theta^\alpha}$ is goal distance in the abstract state space θ^α induced by α. (E.g., Transitions (s, a, s') in the state space yield transitions $(\alpha(s), a, \alpha(s'))$ in θ^α.)

- What is a pattern database heuristic?

  $\rightarrow$ A pattern database heuristic (PDB) is an abstraction heuristic h^α where α is a projection, i.e., $\alpha(s) = \alpha(t)$ iff s and t agree on a subset of the state variables (e.g., those encoding the positions of $1, \dots, 7$ and the blank).

- What is a merge-and-shrink heuristic?

  $\rightarrow$ A merge-and-shrink heuristic (M&S) is an abstraction heuristic h^α conbsructed by starting with projections on single variables, then iteratively merging two abstractions (replacing them with their synchronized product) and shrinking an abstraction (replacing it with an abstraction of itself).

Abstractions: (Some) Recent Results

(a) **Automatic generation of pattern database heuristics [Haslum et al. (2007)]:** The canonical heuristic gets the maximal additive sum from a pattern collection. Find that collection by hill-climbing in the space of pattern collections, pruning useless choices based on the causal graph.

(b) **Merge-and-shrink heuristics [Helmert et al. (2007)]:** Basic framework and a simple instantiation (merging strategy and shrinking strategy).

(c) **Shrinking by bisimulation [Nissim et al. (2011)]:** Bisimulation is a well-known concept from Verification. Using it for shrinking yields perfect heuristics but is prohibitively expensive; conservative label reduction can yield exponential savings at no information loss (happens e.g. in Gripper).

(d) **Shrinking by approximate bisimulation [Katz et al. (2012)]:** To trade accuracy for speed, need coarser notion of state similarity. K-catching bisimulation is bisimulation relative to an action subset K; choosing K enables the trade-off (and is loss-free in certain cases).

$\rightarrow$ None of this is covered in the technical lectures.

The 4 Families (That We Know Up To Now)

Ignoring Deletes

h^{max}
h^+
h^{add}

Abstractions

PDB
M&S

Critical Paths

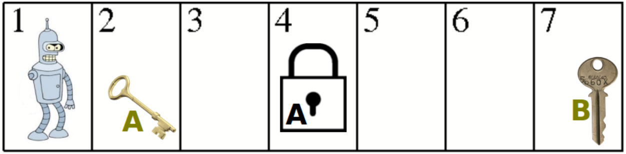
h^1
h^2
h^3
...

Landmarks

h_L^{LM}
(Elementary LM h)

Landmarks: Example

Problem: Bring key B to position 1.



Landmarks:

- robot-at-2, robot-at-3, robot-at-4, robot-at-5, robot-at-6, robot-at-7.
- Lock-open.
- Have-key-A.
- Have-key-B.
- ...

$\rightarrow h =$ **"Number of open items on the to-do list"**

Landmarks: Details

- What is a fact landmark?

  $\rightarrow$ A fact landmark for a state s is a fact that must be true at some point along any plan for s.

- What is a disjunctive action landmark?

  $\rightarrow$ A disjunctive action landmark for a state s is a set of actions L at least one of which must be used by any plan for s.

- How can we turn a fact landmark into a disjunctive action landmark?

  $\rightarrow$ If p is a fact landmark for s, and p is not true in s, then the set L of all actions whose effect includes p is a disjunctive action landmark for s.

- Can *all* disjunctive action landmarks be derived that way?

  $\rightarrow$ No! (Simple counting argument; alternative solution paths.)

- What is the heuristic defined by a disjunctive action landmark L?

  $\rightarrow$ The elementary landmark heuristics h_L^{LM} returns 1 (respectively the cost of the cheapest action in L) if L is a disjunctive action landmark for s, and returns 0 otherwise.

Landmarks: (Some) Recent Results

(a) **Inadmissble LM h for satisficing planning in LAMA [Richter and Westphal (2010)]:** Find fact LMs for initial state, incremental maintenance of open LMs for search states. h =count of open LMs.

(b) **Admissible LM h for optimal planning [Karpas and Domshlak (2009)]:** Find fact LMs for initial state, incremental maintenance of open LMs for search states; consider the induced disjunctive action LMs. h =admissible combination, using cost partitioning (see next section).

(c) **LM-cut [Helmert and Domshlak (2009)]:** Find disjunctive action LMs anew for every search state, using cuts in a graph that defines $h^{\max}$. Combine admissibly using cost partitioning. Empirically, the best admissible heuristic we have at this point!

(d) **From landmarks via hitting sets to h^+ [Bonet and Helmert (2010); Bonet and Castillo (2011); Haslum et al. (2012)]:** Find disjunctive action LMs. Combine admissibly using the minimum-cost hitting set. Given the set of LMs is "complete", this is equal to h^+.

→ (c) and (d), and the cost partitionings of (b), are covered in Lecture 3.

The 4 Families (That We Know Up To Now)

Ignoring Deletes

h^{max}
h^{+}
h^{add}

Abstractions

PDB
M&S

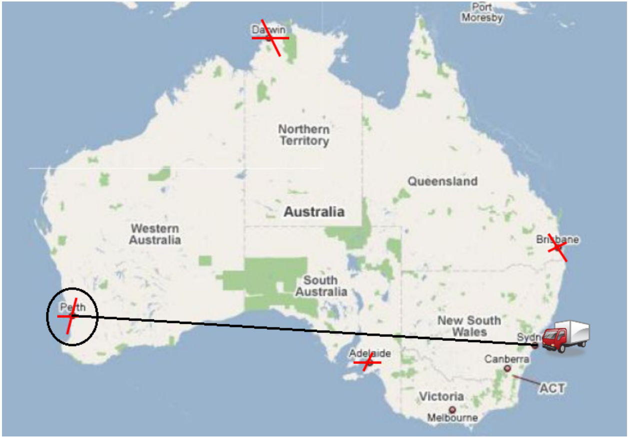
Critical Paths

h^1
h^2
h^3
...

Landmarks

h_L^{LM}
(Elementary LM h)

Critical Paths: Example



$\rightarrow h^1 =$ **Most Expensive** 1**-Sub-Tour**

Critical Paths: Example

$\rightarrow h^2 =$ **Most Expensive 2-Sub-Tour**

Critical Paths: Example

$\rightarrow h^m =$ **Most Expensive m-Sub-Tour**

Critical Paths: Details

- How is h^1 defined?

  $h^1(s) := h^1(s, G)$ where $h^1(s, g)$ is the point-wise greatest function that satisfies $h^1(s, g) =$

$$\begin{cases} 0 & g \subseteq s \\ min_{a \in A, regr(g,a) \text{ is defined}}\, c(a) + h^1(s, regr(g, a)) & |g| = 1 \\ \max_{g' \in g} h^1(s, \{g'\}) & |g| > 1 \end{cases}$$

- Which previously discussed heuristic is h^1 equivalent to? $h^{\max}$.

- How is h^m defined?

  $h^m(s) := h^m(s, G)$ where $h^m(s, g)$ is the point-wise greatest function that satisfies $h^m(s, g) =$

$$\begin{cases} 0 & g \subseteq s \\ min_{a \in A, regr(g,a) \text{ is defined}}\, c(a) + h^m(s, regr(g, a)) & |g| \leq m \\ \max_{g' \subseteq g, |g'| \leq m} h^m(s, g') & |g| > m \end{cases}$$

Critical Paths: (Some) Recent Results

(a) **Compiling h^m into h^1 [Haslum (2009)]:** Given planning task Π, construct a compiled task Π^m such that $h^1(\Pi^m) = h^m(\Pi)$.

(b) **Marriage of h^m with h^+ [Haslum (2012)]:** Given planning task Π, choose a set C of fact conjunctions and construct a compiled task Π^C representing C explicitly, such that:

  (1) If C consists of the size-m conjunctions, then $h^1(\Pi^C) = h^m(\Pi)$.
  (2) $h^+(\Pi) \leq h^+(\Pi^C) \leq h^*(\Pi)$.
  (3) For suitable C, $h^+(\Pi^C) = h^*(\Pi)$.

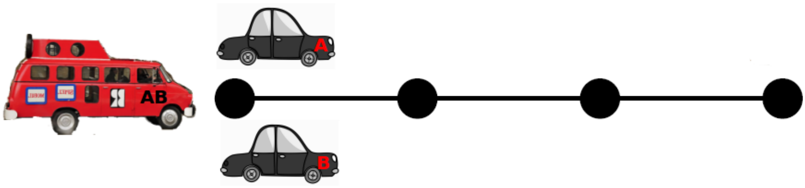
  Unfortunately, $\|\Pi^C\|$ grows exponentially in $|C|$.

(c) **Efficient marriage of h^m with h^+ [Keyder et al. (2012)]:** Given planning task Π, choose a set C of fact conjunctions and construct a compiled task Π^C_{ce} with (1–3), plus being polynomial in $|C|$.

$\rightarrow$ All of (a–c) are (briefly) covered in Lecture 2.

Cost Partitionings: Example

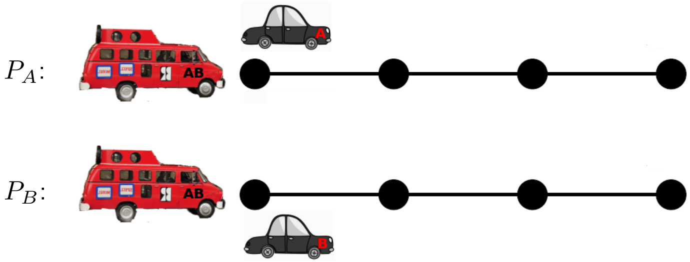
Planning task: Shoot films A and B at the right ($gotA$, $gotB$). "Normal" car A/B can only do film A/B, "fancy" car can do both A and B. Each move of normal car costs 1.5, each move of fancy car costs 2.



Cost Partitionings: Example

Planning task: Shoot films A and B at the right ($gotA$, $gotB$). “Normal” car A/B can only do film A/B, “fancy” car can do both A and B. Each move of normal car costs 1.5, each move of fancy car costs 2.

Heuristics: $P_A = \{carA, fancy, gotA\}$ and $P_B = \{carB, fancy, gotB\}$.



$\rightarrow h^{P_A}(I) = h^{P_B}(I) = 4.5$. $\rightarrow$ Are P_A and P_B additive? No.

Cost partitioning: Normal car X: 1.5 in P_X, 0 in other pattern. Fancy: 1 in each of P_A, P_B. $\rightarrow h^{P_A}(I) = h^{P_B}(I) = 3$ and $h^{P_A}(I) + h^{P_B}(I) = 6 = h^*(I)$.

Cost Partitionings: Details

- How can we always combine admissible heuristics h_1, h_2 into a dominating admissible heuristic? By $\max$.
- Is that a good idea?

  $\rightarrow$ Possibly, but it is much better to take the sum. There are notions of independence for particular families of heuristic functions (e.g., additive PDBs), where independent heuristics can be admissibly summed.
- What is a cost partitioning and why is that useful?

  $\rightarrow$ Given planning task Π and heuristics $h_1, \dots, h_n$ a cost partitioning distributes the cost of each action across n copies Π_i of Π; the partitioned sum then is $\sum_{i=1}^{n} h_i(\Pi_i)$. Cost partitioning applies to any heuristics, and subsumes the earlier notions of independence as special cases.
- How can we find optimal cost partitionings?

  Given a world state s, an optimal cost partitioning is such that the partitioned sum is maximal for s. Such cost partitionings can be found for landmarks and abstractions using Linear Programming (LP).

Cost Partitionings: (Some) Recent Results

(a) **Optimal cost partitionings can be found in polynomial time [Katz and Domshlak (2008)]:** LP-encoding for abstraction heuristics.

(b) **Optimal cost partitionings for elementary LM heuristics [Karpas and Domshlak (2009)]:** LP-encoding for disjunctive action landmarks.

(c) **Targeted practical use of optimal cost partitionings [Karpas et al. (2011)]:** Optimal cost partitionings are great, but they depend on the state. Calling an LP solver for every search state causes enormous runtime overhead. Typically more practical: Solve an LP for some sample states, use a combination/selection of the resulting cost partitionings for each search state.

→ (a) and (b) are covered in Lecture 3.

Compilability Between Admissible Heuristic Functions

Ignoring Deletes

$h^{\mathsf{max}} \leq$
h^{+}

Abstractions

PDB $\preceq$
M&S

Critical Paths

$h^{1} \leq$
$h^{2} \leq$
$h^{3} \leq$
$\ldots$

Landmarks

h^{LM}_{L}
(Elementary LM h)

Ignoring Deletes ↔ Abstractions:
$h^{\mathsf{max}} \not\preceq$ PDB
$h^{\mathsf{max}} \preceq$ M&S

Ignoring Deletes ↔ Critical Paths:
$h^{\mathsf{max}} = h^{1}$

Ignoring Deletes ↔ Landmarks:
$h^{\mathsf{max}} \equiv h^{\mathsf{LM}}_{L}$

Abstractions ↔ Critical Paths:
PDB $\not\preceq h^{m}$
$h^{1} \not\preceq$ PDB
$h^{1} \preceq$ M&S
$h^{m} \not\preceq$ M&S

Abstractions ↔ Landmarks:
$h^{\mathsf{LM}}_{L} \not\preceq$ PDB
$h^{\mathsf{LM}}_{L} \preceq$ M&S
M&S $\not\preceq h^{\mathsf{LM}}_{L}$

Critical Paths ↔ Landmarks:
$h^{1} \equiv h^{\mathsf{LM}}_{L}$

Compilability: Details

- When can a family H of admissible heuristics be compiled into another family H' of admissible heuristics?

  $\rightarrow$ If there exists a polynomial-time algorithm that, given as input any planning task Π, world state s, and $h \in H$, constructs $h'_1, \dots, h'_n \in H'$ along with a cost partitioning so that $h(s) \leq \sum_{i=1}^{n} h'_i(\Pi_i)$.

- So what about partitioned sums of heuristics $h_1, \dots, h_m \in H$?

  $\rightarrow$ Just compile each of them individually, and sum up.

- The standard notion of "heuristic dominance" is the special case where?

  $\rightarrow$ We need only one $h' \in H'$, and the inequality holds for all states s.

Compilability: (Some) Recent Results

(a) **Framework introduction and most proofs [Helmert and Domshlak (2009)]:** Devises the compilability framework, and proves all results stated except $h^m \not\preceq$M&S. Invents LM-cut as a side effect of proving that h^1 can be compiled into elementary LM heuristics. Wonderful paper!

(b) **h^2 (and thus h^m) cannot be compiled into merge-and-shrink:** Soon-to-be-published result proved by Patrik Haslum.

$\rightarrow$ LM-cut (but nothing else of this) is covered in Lecture 3.

Summary

- Heuristic search is a prominent approach to planning, with lots of success in the IPC (and also, successes in applications).
- A key question is how to generate the goal distance estimator, i.e., the heuristic function h.
- The investigation of that question is reasonably mature, with 4 families of methods, along with general techniques for combining and comparing these methods.
- Many exciting results have been published very recently. Some of them are covered in detail in the following two technical lectures.
- For timing reasons, the technical lectures have to (a) assume a basic familiarity with the subject, and (b) gloss over/omit many details.

  To counter (b), following the citations is strongly recommended!

  If you don't qualify for (a), you can have a look at my lecture slides, which introduce all the basic concepts in detail:

  `http://fai.cs.uni-saarland.de/teaching/winter12-13/planning.html`

References I

Blai Bonet and Julio Castillo. A complete algorithm for generating landmarks. In Fahiem Bacchus, Carmel Domshlak, Stefan Edelkamp, and Malte Helmert, editors, *Proceedings of the 21st International Conference on Automated Planning and Scheduling (ICAPS 2011)*. AAAI Press, 2011.

Blai Bonet and Malte Helmert. Strengthening landmark heuristics via hitting sets. In Helder Coelho, Rudi Studer, and Michael Wooldridge, editors, *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI'10)*, pages 329–334, Lisbon, Portugal, August 2010. IOS Press.

Patrik Haslum, Adi Botea, Malte Helmert, Blai Bonet, and Sven Koenig. Domain-independent construction of pattern database heuristics for cost-optimal planning. In Adele Howe and Robert C. Holte, editors, *Proceedings of the 22nd National Conference of the American Association for Artificial Intelligence (AAAI-07)*, pages 1007–1012, Vancouver, BC, Canada, July 2007. AAAI Press.

Patrik Haslum, John Slaney, and Sylvie Thiébaux. Minimal landmarks for optimal delete-free planning. In Blai Bonet, Lee McCluskey, José Reinaldo Silva, and Brian Williams, editors, *Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS 2012)*, pages 353–357. AAAI Press, 2012.

References II

Patrik Haslum. $h^m(P) = h^1(P^m)$: Alternative characterisations of the generalisation from $h^{\max}$ to h^m. In Alfonso Gerevini, Adele Howe, Amedeo Cesta, and Ioannis Refanidis, editors, *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS 2009)*, pages 354–357. AAAI Press, 2009.

Patrik Haslum. Incremental lower bounds for additive cost planning problems. In Blai Bonet, Lee McCluskey, José Reinaldo Silva, and Brian Williams, editors, *Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS 2012)*, pages 74–82. AAAI Press, 2012.

Malte Helmert and Carmel Domshlak. Landmarks, critical paths and abstractions: What's the difference anyway? In Alfonso Gerevini, Adele Howe, Amedeo Cesta, and Ioannis Refanidis, editors, *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS 2009)*, pages 162–169. AAAI Press, 2009.

Malte Helmert, Patrik Haslum, and Jörg Hoffmann. Flexible abstraction heuristics for optimal sequential planning. In Mark Boddy, Maria Fox, and Sylvie Thiebaux, editors, *Proceedings of the 17th International Conference on Automated Planning and Scheduling (ICAPS-07)*, pages 176–183, Providence, Rhode Island, USA, 2007. Morgan Kaufmann.

References III

Jörg Hoffmann. Analyzing search topology without running any search: On the connection between causal graphs and h^+. *Journal of Artificial Intelligence Research*, 41:155–229, 2011.

Erez Karpas and Carmel Domshlak. Cost-optimal planning with landmarks. In C. Boutilier, editor, *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009)*, pages 1728–1733, Pasadena, California, USA, July 2009. Morgan Kaufmann.

Erez Karpas, Michael Katz, and Shaul Markovitch. When optimal is just not good enough: Learning fast informative action cost-partitionings. In Fahiem Bacchus, Carmel Domshlak, Stefan Edelkamp, and Malte Helmert, editors, *Proceedings of the 21st International Conference on Automated Planning and Scheduling (ICAPS 2011)*, pages 122–129. AAAI Press, 2011.

Michael Katz and Carmel Domshlak. Optimal additive composition of abstraction-based admissible heuristics. In Jussi Rintanen, Bernhard Nebel, J. Christopher Beck, and Eric Hansen, editors, *Proceedings of the 18th International Conference on Automated Planning and Scheduling (ICAPS 2008)*, pages 174–181. AAAI Press, 2008.

References IV

Michael Katz, Jörg Hoffmann, and Malte Helmert. How to relax a bisimulation? In Blai Bonet, Lee McCluskey, José Reinaldo Silva, and Brian Williams, editors, *Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS 2012)*. AAAI Press, 2012.

Michael Katz, Jörg Hoffmann, and Carmel Domshlak. Red-black relaxed plan heuristics. In Marie desJardins and Michael Littman, editors, *Proceedings of the 27th National Conference of the American Association for Artificial Intelligence (AAAI'13)*, Bellevue, WA, USA, July 2013. AAAI Press. Forthcoming.

Michael Katz, Jörg Hoffmann, and Carmel Domshlak. Who said we need to relax *All* variables? In Daniel Borrajo, Simone Fratini, Subbarao Kambhampati, and Angelo Oddi, editors, *Proceedings of the 23rd International Conference on Automated Planning and Scheduling (ICAPS 2013)*. AAAI Press, 2013. Forthcoming.

Emil Keyder, Jörg Hoffmann, and Patrik Haslum. Semi-relaxed plan heuristics. In Blai Bonet, Lee McCluskey, José Reinaldo Silva, and Brian Williams, editors, *Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS 2012)*. AAAI Press, 2012.

References V

Raz Nissim, Jörg Hoffmann, and Malte Helmert. Computing perfect heuristics in polynomial time: On bisimulation and merge-and-shrink abstraction in optimal planning. In Toby Walsh, editor, *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*, pages 1983–1990. AAAI Press/IJCAI, 2011.

Silvia Richter and Matthias Westphal. The LAMA planner: Guiding cost-based anytime planning with landmarks. *Journal of Artificial Intelligence Research*, 39:127–177, 2010.