LM-Cut: Optimal Planning with the Landmark-Cut Heuristic*

Malte Helmert
Albert-Ludwigs-Universität Freiburg, Germany
helmert@informatik.uni-freiburg.de

Carmel Domshlak
Technion, Israel
dcarmel@ie.technion.ac.il

Abstract

The LM-Cut planner uses the landmark-cut heuristic, introduced by the authors in 2009, within a standard A* progression search framework to find optimal sequential plans for STRIPS-style planning tasks. This short paper recapitulates the main ideas surrounding the landmark-cut heuristic and provides pointers for further reading.

Introduction

Heuristic search, either in the space of world states reached through progression or in the space of subgoals reached through regression, is a common and successful approach to classical planning. Apart from the choice of search algorithm, the main feature that distinguishes heuristic planners is their heuristic estimator. Most current heuristic functions are based on one of the following four ideas:

1. *delete relaxation*: e. g., h^+ (Hoffmann and Nebel 2001), h^{max} (Bonet and Geffner 2001), h^{add} (Bonet and Geffner 2001), h^{FF} (Hoffmann and Nebel 2001), h^{pmax} (Mirkis and Domshlak 2007), h^{sa} (Keyder and Geffner 2008)
2. *critical paths*: the h^m heuristic family (Haslum and Geffner 2000)
3. *abstraction*: pattern databases (Edelkamp 2001), merge-and-shrink abstractions (Helmert, Haslum, and Hoffmann 2007), and structural patterns (Katz and Domshlak 2008)
4. *landmarks*: LAMA's h^{LM} (Richter, Helmert, and Westphal 2008; Richter and Westphal 2010) and the admissible landmark heuristics h^{L} and h^{LA} (Karpas and Domshlak 2009)

These ideas have been developed in relative isolation. For a long time, apart from Haslum and Geffner's (2000) result that h^{max} is a special case of the h^m family ($h^{\text{max}} = h^1$), no formal connections between these different ideas for devising heuristic estimators had been known. In a recent paper (Helmert and Domshlak 2009), we addressed this issue by proving a number of *dominance results*, which established, subject to the usual complexity-theoretic assumption that polynomial overhead is acceptable, the following relationships:

- Landmark heuristics dominate additive h^{max} heuristics.
- Additive h^{max} heuristics dominate landmark heuristics.
- Additive critical path heuristics with $m \geq 2$ strictly dominate landmark heuristics and additive h^{max} heuristics.
- Merge-and-shrink abstraction heuristics strictly dominate landmark heuristics and additive h^{max} heuristics.
- Pattern database heuristics are incomparable with landmark heuristics and additive h^{max} heuristics.

These statements are informal summaries, and some restrictions apply. In particular, the results for landmark heuristics only apply to *relaxation-based* landmarks, which are verifiable by a relaxed planning graph criterion. Until very recently, all landmark heuristics in the literature fell into this class. However, this has changed with the work of Keyder, Richter, and Helmert (2010), who introduced landmarks based on the h^m heuristic family.

On the positive side, all dominance results are constructive, showing how to compute a dominating heuristic in polynomial time. Moreover, some of the compilations are efficient enough to be worth implementing in practice. We implemented one such construction, from the regular (non-additive) h^{max} heuristic to landmarks, to obtain a new heuristic, which we called the *landmark-cut heuristic* $h^{\text{LM-cut}}$.

The Landmark-Cut Heuristic

The landmark-cut heuristic can alternatively be viewed as a landmark heuristic, a cost-partitioning scheme for additive h^{max}, or an approximation to the (intractable) optimal relaxation heuristic h^+.

Here, we briefly recapitulate the computation of $h^{\text{LM-cut}}$. We assume familiarity with fundamental concepts such as delete relaxation, landmarks, and the h^{max} and h^+ heuristics. For readers who are new to these concepts, we refer to our original paper on $h^{\text{LM-cut}}$ (Helmert and Domshlak 2009) and the later work by Bonet and Helmert (2010), which related $h^{\text{LM-cut}}$ to hitting sets and showed that a generalization of $h^{\text{LM-cut}}$ based on hitting sets always achieves the perfect delete relaxation estimate h^+ when allowed exponential computation time.

To determine the $h^{\text{LM-cut}}$ estimate of a state s, we first compute $h^{\text{max}}(s)$. If this value is zero or infinite, this implies that $h^+(s)$ is also zero or infinite, respectively, and hence

*Our presentation in this paper borrows heavily from the earlier paper in which we introduced the landmark-cut heuristic (Helmert and Domshlak 2009).

this is the best possible information that can be extracted from the delete relaxation of the task. In these cases, we set $h^{\text{LM-cut}}(s) = h^{\max}(s)$.

Otherwise, the cost to solve the delete relaxation of the given task from state s is finite and nonzero. In this case, we compute a *nontrivial disjunctive action landmark* of the delete relaxation, which is a set L of actions of nonzero cost such that each relaxed plan solving the task from the given state must include an action from L.

After computing such a landmark, we add the minimal cost c among all actions in L to the heuristic value computed so far (which is initialized to 0), reduce the cost of all actions in L by c, and then start again by recomputing the $h^{\max}$ values based on the reduced action costs, computing a new disjunctive action landmark, and so on. The process ends once action costs have been reduced to the extent that the $h^{\max}$ estimate of the resulting problem becomes zero.

The main challenge in this computation is finding a suitable landmark L. It is not particularly hard to find *some* such landmark: the set of all actions of nonzero cost will do. However, the larger the set L, the more actions will have their cost reduced at the end of the current iteration of the main landmark-cut loop, leading to potentially fewer landmarks that can be extracted in future rounds. The challenge, then, is in finding a reasonably small such set.

The landmark-cut heuristic addresses this issue by computing so-called *justification graphs*, which "justify" the $h^{\max}$ values of the facts of the planning task by linking each effect of an action a to the most expensive precondition of a (or one of the most expensive ones, in case of ties).

Arcs in justification graphs are weighted by the costs of the actions that induce them. A shortest paths in a justification graph corresponds to a causal chain whose cost explains the $h^{\max}$ value of a fact, and *cuts* in justification graphs (sets of arcs whose removal disconnects the current state from the goal) correspond to disjunctive action landmarks. These relationships are explored in more depth by Bonet and Helmert (2010), who show that *all* relevant landmarks of the delete relaxation can be computed as cuts in justification graphs when arbitrary preconditions are allowed to induce arcs in the graph (rather than just preconditions with maximal $h^{\max}$ value).

The landmark selected by the landmark-cut heuristic is based on a particular cut close to the goal facts of the task, which is sufficient to guarantee that the final heuristic value is always at least as large as $h^{\max}(s)$. (In our experiments, it is usually much larger.)

The LM-Cut Planner

In our earlier work (Helmert and Domshlak 2009), we demonstrated experimentally that $h^{\text{LM-cut}}$ gives excellent approximations to h^+ and compares favourably to other admissible planning heuristics in terms of accuracy. We also showed that an optimal planner based on A^* search with the landmark-cut heuristic was highly competitive with the state of the art at the time.

The LM-Cut planner entered into IPC 2011 is almost identical to the system used in these experiments. The only two changes since then are:

- minor bug fixes and performance improvements in various components of the Fast Downward planner that serves as the basis of our implementation, and
- support for actions of non-unit cost. (While our original description of the landmark-cut heuristic was fully general, our implementation was restricted to the unit-cost case.)

References

Bonet, B., and Geffner, H. 2001. Planning as heuristic search. *Artificial Intelligence* 129(1):5–33.

Bonet, B., and Helmert, M. 2010. Strengthening landmark heuristics via hitting sets. In Coelho, H.; Studer, R.; and Wooldridge, M., eds., *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010)*, 329–334. IOS Press.

Edelkamp, S. 2001. Planning with pattern databases. In Cesta, A., and Borrajo, D., eds., *Pre-proceedings of the Sixth European Conference on Planning (ECP 2001)*, 13–24.

Haslum, P., and Geffner, H. 2000. Admissible heuristics for optimal planning. In Chien, S.; Kambhampati, S.; and Knoblock, C. A., eds., *Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling (AIPS 2000)*, 140–149. AAAI Press.

Helmert, M., and Domshlak, C. 2009. Landmarks, critical paths and abstractions: What's the difference anyway? In Gerevini, A.; Howe, A.; Cesta, A.; and Refanidis, I., eds., *Proceedings of the Nineteenth International Conference on Automated Planning and Scheduling (ICAPS 2009)*, 162–169. AAAI Press.

Helmert, M.; Haslum, P.; and Hoffmann, J. 2007. Flexible abstraction heuristics for optimal sequential planning. In Boddy, M.; Fox, M.; and Thiébaux, S., eds., *Proceedings of the Seventeenth International Conference on Automated Planning and Scheduling (ICAPS 2007)*, 176–183. AAAI Press.

Hoffmann, J., and Nebel, B. 2001. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research* 14:253–302.

Karpas, E., and Domshlak, C. 2009. Cost-optimal planning with landmarks. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009)*, 1728–1733.

Katz, M., and Domshlak, C. 2008. Structural patterns heuristics via fork decomposition. In Rintanen, J.; Nebel, B.; Beck, J. C.; and Hansen, E., eds., *Proceedings of the Eighteenth International Conference on Automated Planning and Scheduling (ICAPS 2008)*, 182–189. AAAI Press.

Keyder, E., and Geffner, H. 2008. Heuristics for planning with action costs revisited. In *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI 2008)*, 588–592.

Keyder, E.; Richter, S.; and Helmert, M. 2010. Sound and complete landmarks for and/or graphs. In Coelho, H.; Studer, R.; and Wooldridge, M., eds., *Proceedings of the*

19th European Conference on Artificial Intelligence (ECAI 2010), 335–340. IOS Press.

Mirkis, V., and Domshlak, C. 2007. Cost-sharing approximations for h^+. In Boddy, M.; Fox, M.; and Thiébaux, S., eds., *Proceedings of the Seventeenth International Conference on Automated Planning and Scheduling (ICAPS 2007)*, 240–247. AAAI Press.

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

Richter, S.; Helmert, M.; and Westphal, M. 2008. Landmarks revisited. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI 2008)*, 975–982. AAAI Press.