# Landmark-based Meta Best-First Search

Bachelor Thesis Presentation

By Samuel Hugger

# STRIPS Planning Model

- States
  - Set of atoms
    - True or False
- Actions
  - Preconditions
  - Add-effects
  - Delete-effects

# STRIPS Planning Model

- Planning Task
  - States
  - Actions
  - Initial state
  - Goal condition

- Solution plan
  - Sequence of actions from initial state to goal

# Landmarks

- Landmarks are facts that have to be true at some point of every solution plan → necessary conditions for reaching any goal

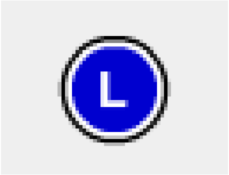
- We consider causal landmarks which correspond to atoms (Zhu & Givan, 2003)

# Landmark-based Search

- Successful: landmarks in heuristic functions
  - LM-count heuristic
  - LM-cut heuristic
- Not as successful: landmarks in meta-search – lack of completeness
- → Landmark-based Meta Best-First Search Algorithm (LMBFS), Vernhes et al., 2013

# Landmark-based Meta-Search

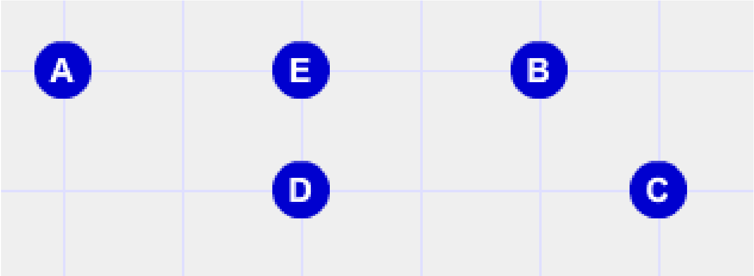
- The idea: Divide the planning task into subtasks
- Each subtask's goal is the achievement of a landmark




- Subtask ordering?

# Landmark ordering

- A bunch of landmarks




- Order them in a way that is benefitial to reaching the goal of the planning task

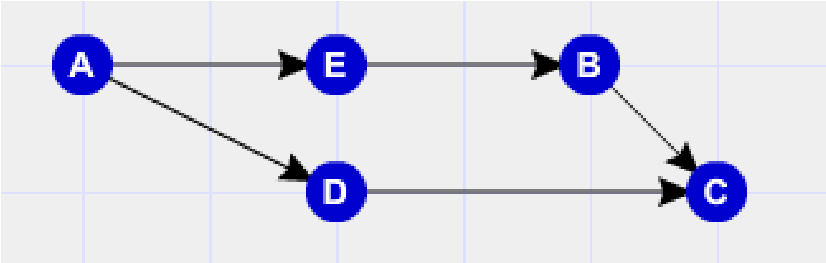
# Precedence Relation

**Definition 2.** (Precedence relation $<_{\mathcal{L}}$). [1]
$<_{\mathcal{L}}$ can be defined on a set of landmarks $\mathcal{L}$. For two landmarks $(l, l') \in \mathcal{L}^2$, $l <_{\mathcal{L}} l'$ if $l$ becomes true before $l'$ becomes true during the execution of every solution plan.

- We order our landmarks based on the precedence relation.

# Landmark graph

- The resulting landmark graph is oriented towards solution plans



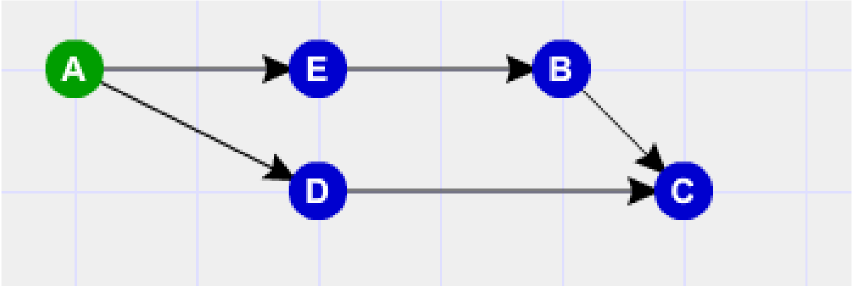

- Good starting points for the search:

**Definition 5.** (Root landmark set). [1]

Let $\Gamma = (\mathcal{V}, \mathcal{E})$ be a landmark graph. We define $roots(\Gamma) = \{l \in \mathcal{V} \mid Pa_\Gamma(l) = \emptyset\}$.

# Landmark graph

- Root landmarks in this graph:




- Node A is a root landmark – it is likely to be achieved early in every solution plan

# Metanodes

**Definition 7.** (Metanode). [1]
A *metanode* is a tuple $m = \langle s, h, \mathcal{A}, l, \rho \rangle$ where:

- $s$ is a state of the planning task $\Pi$
- $h$ is a heuristic evaluation of the node
- $\mathcal{A}$ is a set of landmarks ($\mathcal{A} \subseteq \mathcal{L}$)
- $l$ is a landmark ($l \in \mathcal{L}$)
- $\rho$ is a plan yielding the state $s$ from the initial state $I$.

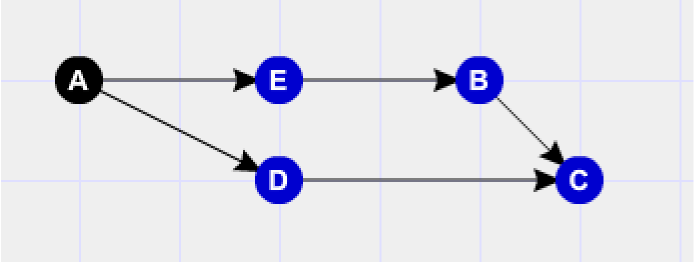
- The subtask associated to a metanode m has the landmark l as goal

# Subtask action restriction

- Actions must either or:
  - achieve l
  - not achieve any root landmarks
- This focuses the subsearch on l
- → Run subsearch – if successful, expand the associated metanode

# Expansion of Metanodes

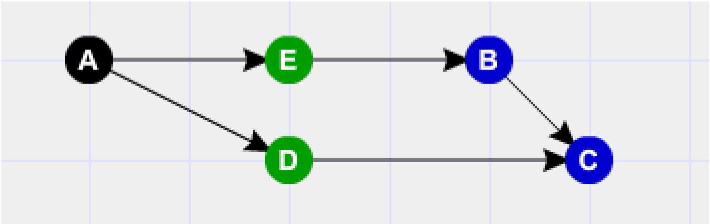
- Achieved landmarks are removed from the landmark graph




- New metanodes are generated and added to the open list

# Metanode Generation - nextLM

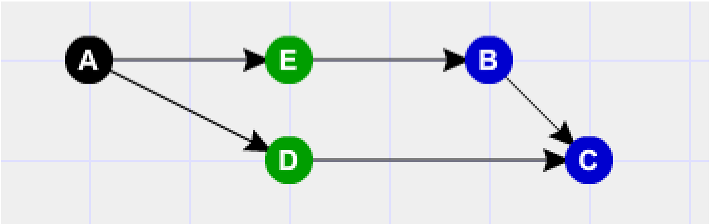
**Definition 9.** (nextLM metanode generation). [1]
Let $m = \langle s, h, \mathcal{A}, l, \rho \rangle$ be a metanode. If $\Pi_m$ has a solution $\rho'$, then $nextLM(m) = \{\langle s', h', \mathcal{A} \cup \{l\}, l', (\rho \circ \rho') \rangle \mid l' \in roots(\Gamma \backslash (\mathcal{A} \cup \{l\}))\}$. If $\Pi_m$ has no solution, then $nextLM(m) = \emptyset$.




- A – expanded metanode
- E, D – generated metanodes

# Metanode Generation - deleteLM

**Definition 10.** (deleteLM metanode generation). [1]
Let $m = \langle s, h, \mathcal{A}, l, \rho \rangle$ be a metanode. $nextLM(m) = \{\langle s, h', \mathcal{A} \cup \{l\}, l', \rho \rangle \mid l' \in roots(\Gamma \setminus (\mathcal{A} \cup \{l\}))\}$.




- No subsearch is run on A – A is removed from the landmark graph
- E,D – generated metanodes

# Completeness

- $succDel(m) = nextLM(m) \cup deleteLM(m)$
- $succCut(m) = nextLM(m) \cup cutParents(m) \cup restartCutParents(m)$

**Theorem 1.** *[1]*
*The LMBFS algorithm using succCut or succDel as successor function is sound and complete if the subplanner is sound and complete.*

# Best-first Search - Heuristics

- LMBFS uses heuristics to select the most promising metanode in each iteration

**Definition 13.** $(h^{\mathcal{L}_{left}})$ [1]
For a metanode $m = \langle s, h, \mathcal{A}, l, \rho \rangle$ and an associated landmark graph $\Gamma = (\mathcal{V}, \mathcal{E})$, the heuristic $h^{\mathcal{L}_{left}}(m) = |\mathcal{V} \backslash \mathcal{A}|$.

- This heuristic works well with LMBFS, as the set of achieved landmarks is already saved in each metanode

# The LMBFS Algorithm

**Algorithm 1:** LMBFS [1]

**Input :** STRIPS planning task $\Pi = \langle A, O, I, G \rangle$, landmark graph $\Gamma$, metanode successor function *succ*

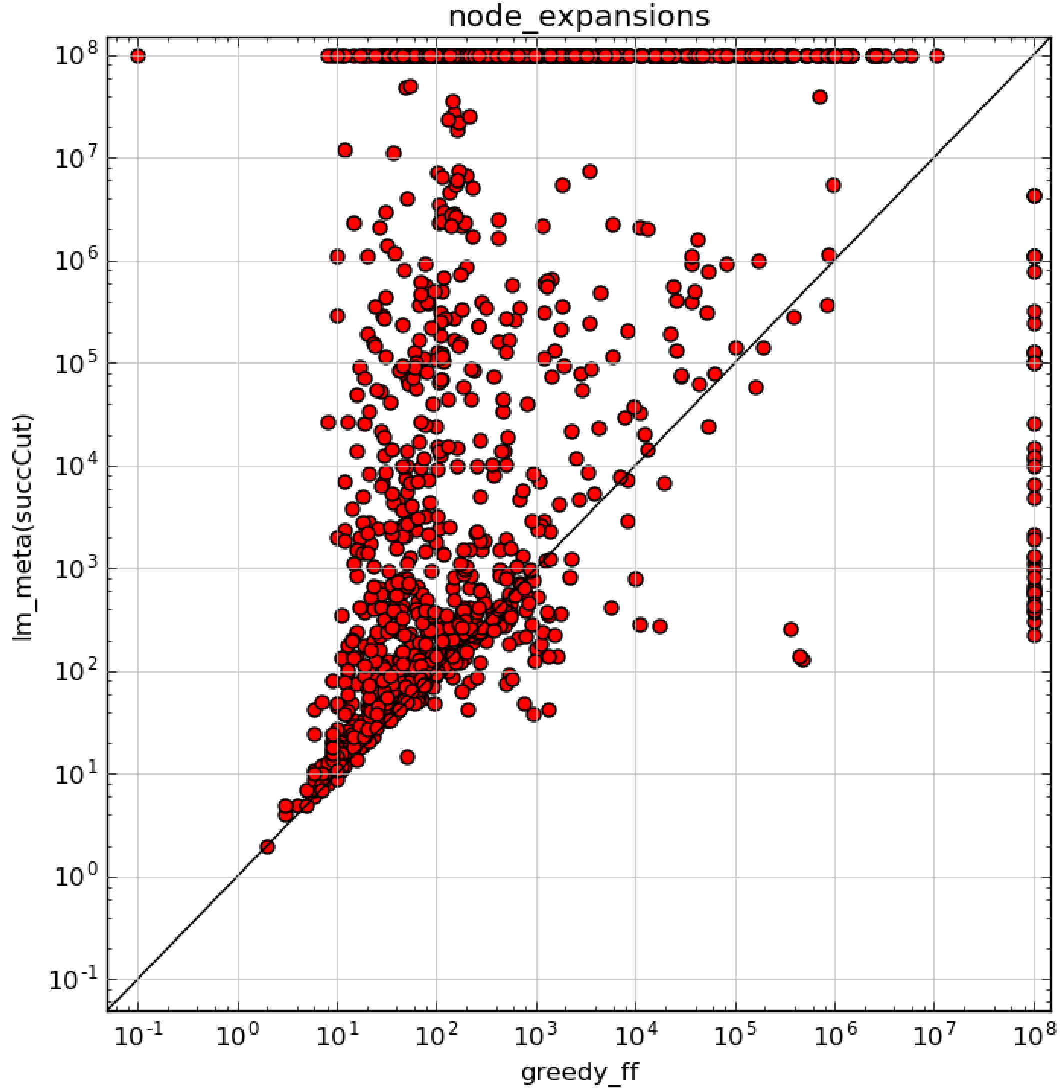
**Output:** solution plan or $\bot$

$open \leftarrow \emptyset$;
$closed \leftarrow \emptyset$;
$\forall l \in roots(\Gamma)$ : add metanode $\langle I, h, \emptyset, l, \emptyset \rangle$ to $open$;
**while** open $\neq \emptyset$ **do**
  $m \leftarrow \arg\min_{\langle s,h,\mathcal{A},l,\rho \rangle \in open} h$;
  $open \leftarrow open \setminus \{m\}$;
  **if** $m \notin closed$ **then**
    $closed \leftarrow closed \cup \{m\}$;
    $\rho' \leftarrow subplanner(\Pi_m)$;
    **if** $\rho' \neq \bot$ **then**
      $s' \leftarrow$ result of executing $\rho'$ in $s$;
      **if** $G \subset s'$ **then**
        **return** $\rho \circ \rho'$;
      $open \leftarrow open \cup succ(m)$;
**return** $\bot$

# LMBFS Evaluation

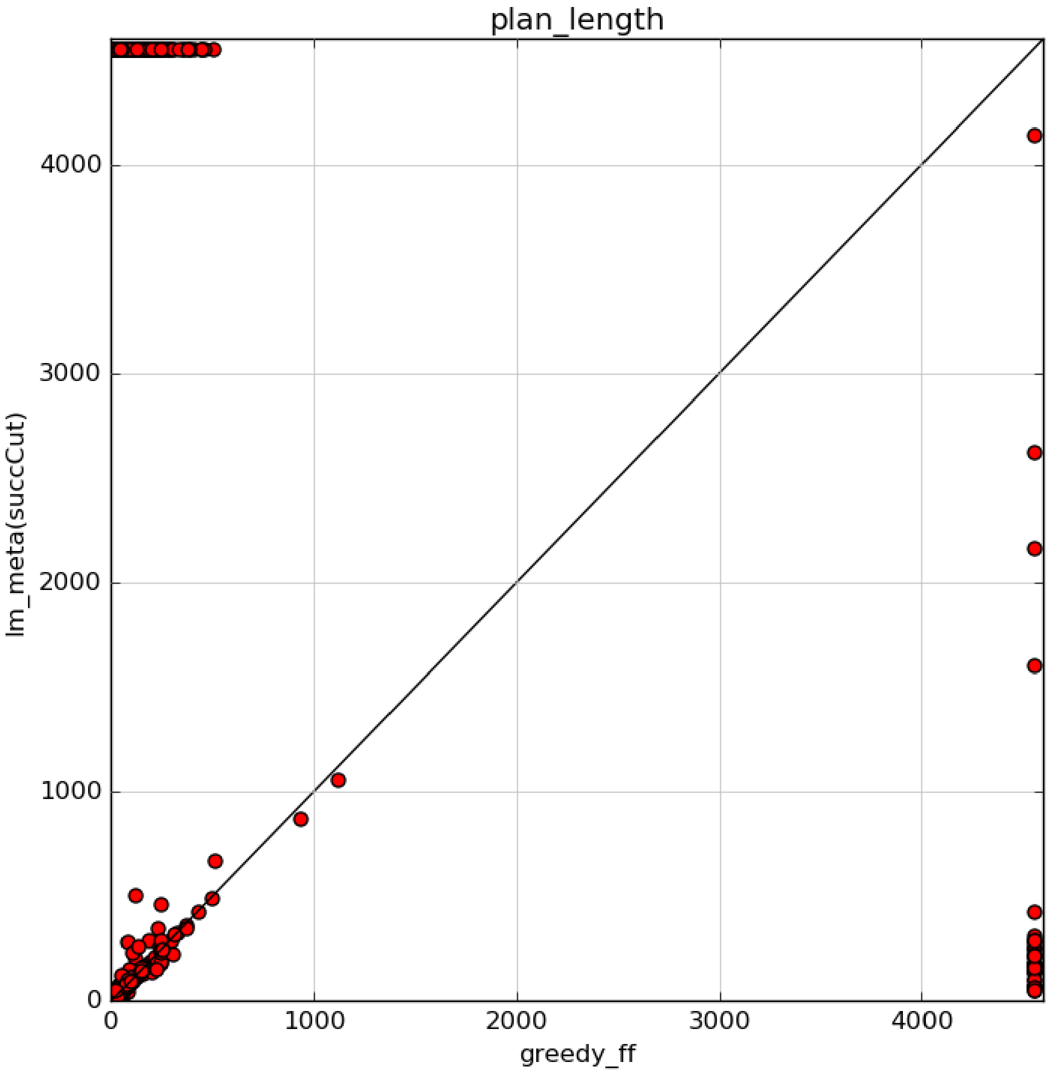
- LMBFS has been implemented in Fast Downward
- Eager-Greedy search as subplanner
- Eager-Greedy search for comparison
- Experiments have been run on the Maia Cluster, using downward-lab

# LMBFS Evaluation



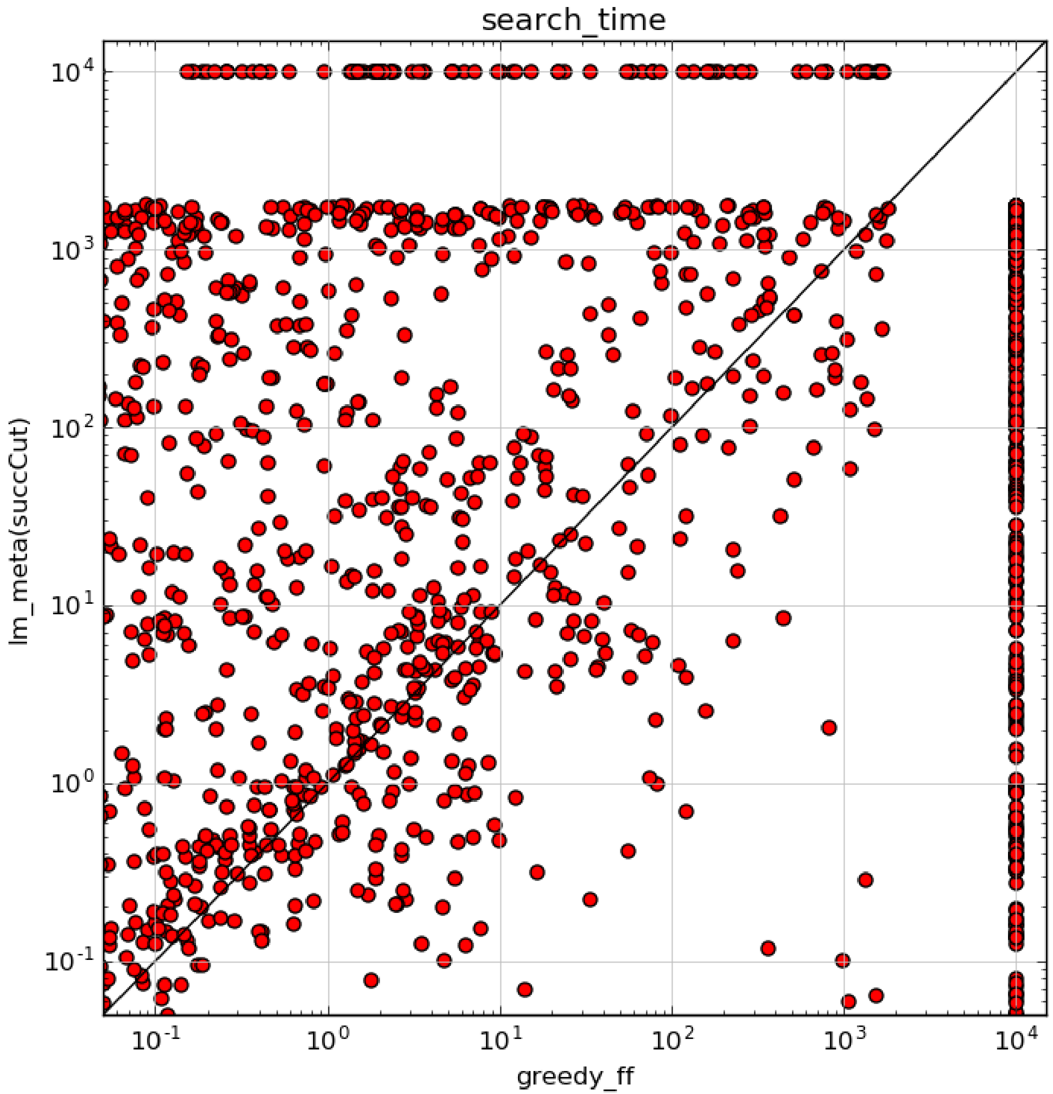

| greedy_ff | lm_meta(succCut) |
|---|---|
| 147.76 | 1527.58 |

# LMBFS Evaluation




| greedy_ff | lm_meta(succCut) |
|---|---|
| **32469** | 35848 |

# LMBFS Evaluation




| greedy_ff | lm_meta(succCut) |
|---|---|
| 0.65 | 3.97 |

# LMBFS Evaluation

| **Summary** | **greedy_ff** | | **lm_meta(succCut)** | **lm_meta(succDel)** |
|---|---|---|---|---|
| Cost - Sum | **5784772.00** | 1 | 5794109.00 | 5794379.00 |
| Coverage - Sum | **1438** | 0.6 | 878 | 801 |
| Evaluated - Sum | **30785444** | 10 | 318829794 | 927036208 |
| Expansions - Geometric mean | **147.76** | 10 | 1527.58 | 3729.37 |
| Generated - Geometric mean | **1567.10** | 6.6 | 10317.14 | 24509.36 |
| Plan length - Sum | **32469** | 1.1 | 35848 | 36605 |
| Search time - Geometric mean | **0.65** | 6.1 | 3.97 | 8.92 |

This table shows a summary of the experiments, aggregated across all domains.

# LMBFS Evaluation

- This implementation of LMBFS is at least a few optimizations away from being competitive with EagerGreedy search
- LMBFS implemented by Vernhes et al. in 2003 has been shown to be competitive with the LAMA-11 planner on 14 IPC-domains

# Conclusion

- Landmark-based Meta Best-First Search represents a successful realization of landmarks as an effective tool in a meta-search environment
- Meta-search is a highly flexible framework with a number of unexplored areas – new successor functions, dynamic successor function choice, the interplay of meta-search and subsearch, and many more

# Bibliography

[1] Vernhes, S., Infantes, G., and Vidal, V. Landmark-based Meta Best-First Search Algorithm: First Parallelization Attempt and Evaluation. In *In Proceedings of the ICAPS-2013 Workshop on Heuristics and Search for Domain-independent Planning (HSDIP 2013)* (2013).

[2] Helmert, M. The Fast Downward Planning System. In *Journal of Artificial Intelligence Research 26 (2006), 191-246* (2006).

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

[4] Helmert, M. and Domshlak, C. Landmarks, Critical Paths and Abstractions: What's the Difference Anyways? In *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS 2009), pp. 162-169. 2009* (2009).

[5] Bonet, B. and Geffner, H. Planning as heuristic search. In *Artificial Intelligence 129 (2001), 5-33* (2001).

[6] Hoffmann, J. and Nebel, B. The FF Planning System: Fast Plan generation through heuristic search. In *Journal of Artificial Intelligence Research, 14 (2001), 253-302* (2001).

[7] Zhu, L. and Givan, R. Landmark extraction via planning graph propagation. In *Proceedings of the ICAPS'03 Doctoral Consortium, 156-160.* (2003).

[8] Richter, S. and Helmert, M. Preferred operators and deferred evaluation in satisficing planning. In *Proceedings of the Nineteenth International Conference on Automated Planning and Scheduling (ICAPS 2009), pp. 273-280. AAAI Press* (2009).

[9] Seipp, J., Pommerening, F., Sievers, S., and Helmert, M. downward-lab 2.0 (2017). URL https://doi.org/10.5281/zenodo.399255.