Variants of Enforced Hill Climbing

Eda Kaynar

BSc Thesis Presentation

Examiner: Prof. Dr. Malte Helmert
Supervisor: Dr. Clemens Büchner

17. July 2026

Planning, informally

A **state**: what's currently true (e.g. "I'm at home", "shoes off").

An **action**: changes some of these facts (e.g. "put shoes on").

A **goal**: some facts we want to become true ("I'm in the seminar room").

A **plan**: a sequence of actions from the start state to a goal state.

start $\longrightarrow \bullet \longrightarrow \bullet \longrightarrow \bullet \longrightarrow$ **goal**

How do we search this space of states efficiently?

Hill Climbing

Always move to a neighbouring state with a **strictly lower** h.

[board example]

Enforced Hill Climbing

If no neighbour improves: run **Breadth-First Search** outward until one does.

[board example, continued]

Fast Forward vs. Fast Downward

Fast Forward (FF) (2001)

- Local closed list (reset each phase)
- Non-lazy heuristic evaluation
- Helpful actions (**prune**/rank)

Fast Downward (FD) (2006)

- Global closed list (never reset)
- Lazy heuristic evaluation
- Preferred operators (prune/**rank**)

Same algorithm, same heuristic; three independent implementation choices differ.

1. Global vs. Local Closed List

Can a **closed** state be revisited?

FD (global): never, closed states stay closed forever.

FF (local): yes, the closed list resets every phase.

[board example, continued]

2. Lazy vs. Non-Lazy Heuristic Evaluation

When is h computed: at **generation** time, or at **expansion** (pop) time?

Non-lazy (FF): compute h immediately for every successor.

Lazy (FD): defer h until the state is popped for expansion.

[board example, continued]

3. Preferred Operators vs. Helpful Actions

Both come from the **relaxed plan** (ignore delete effects).

Helpful action: action that achieves at least one fact the relaxed plan says it needs **next**

Preferred operator: an action the relaxed plan actually **picked** at this step, a stricter subset.

[board example]

Prune or Rank

Prune (`PRUNE_BY_PREFERRED`, original FF): **only** the selected actions ever enter the open list. Fast, but can throw away the only path to the goal.

Rank (`RANK_PREFERRED_FIRST`, FD): **all** successors are kept, selected ones just go first. Slower per phase, never discards a successor outright.

Both helpful actions and preferred operators can be combined with either mode.

Extension: Dead-End Pruning

New `dead_end_list`: remembers states with $h = \infty$.

Skip them immediately if seen again, avoids recomputing h for known dead ends.

Most useful with a **local** closed list, which forgets everything else every phase.

Optional, off by default.

Result 1: Local vs. Global Closed List

Metric	Local	Global
Cost (sum)	**8,955,381**	8,963,266
Coverage	1,037	**1,279**
Unsolved-inc.	**487**	574
Evals (geo mean)	3,337	**715**
Expansions (geo mean)	3,296	**562**
Generated (geo mean)	10,734	**1,431**
Total time (geo mean)	0.57 s	**0.20 s**

Fixed: `global`

Result 2: Lazy vs. Non-Lazy

Metric	Non-lazy	Lazy
Cost (sum)	**9,074,728**	11,294,933
Coverage	1,031	**1,279**
Unsolved-inc.	**566**	574
Evals (geo mean)	4,595	**779**
Expansions (geo mean)	1,148	**605**
Generated (geo mean)	12,461	**1,591**
Total time (geo mean)	0.54 s	**0.19 s**

Fixed: `lazy`

Result 3: Preferred Ops vs. Helpful Actions

Metric	Pref. ops	Helpful actions
Cost (sum)	8,958,776	**8,958,464**
Coverage	**1,279**	1,264
Unsolved-inc.	**474**	479
Evals (geo mean)	713	**698**
Expansions (geo mean)	554	**542**
Generated (geo mean)	1,442	**1,409**
Total time (geo mean)	**0.17 s**	0.19 s

Preferred operators vs. helpful actions under `RANK_PREFERRED_FIRST`, with `l=1,gc=1` fixed.

Result 3: Preferred Ops vs. Helpful Actions

Metric	Pref. ops	Helpful actions
Cost (sum)	8,952,213	**8,951,933**
Coverage	**1,279**	1,266
Unsolved-inc.	482	**479**
Evals (geo mean)	716	**693**
Expansions (geo mean)	557	**538**
Generated (geo mean)	1,449	**1,396**
Total time (geo mean)	0.17 s	0.17 s

Preferred operators vs. helpful actions under `PRUNE_BY_PREFERRED`, with `l=1,gc=1` fixed.

Best and Worst Overall Configurations

Configuration	Description	Coverage	Time	Unsolved-inc.
l=1,gc=1,ha=0,de=0	Fast Downward	**1,279**	**0.17 s**	482
l=1,gc=1,ha=1,de=1	Hybrid	1,264	0.19 s	**479**
l=0,gc=0,ha=1	Fast Forward-style	999	1.28 s	479
l=0,gc=0,ha=0,de=0	Worst overall	1,002	1.07 s	481

FD's own defaults win overall on coverage and speed.

No configuration reaches 0 unsolved-incomplete; EHC is incomplete.

Conclusion and Outlook

FF and FD run the **same** algorithm but diverge in three independent choices, each with a real, sometimes opposing effect.

Global closed list: big efficiency win, genuine extra incompleteness.

Lazy evaluation: biggest efficiency gain.

Prune vs. rank mostly changes **how** EHC fails, not **whether**.

FD's defaults are well-calibrated for this benchmark set.

Compare the actual **plans** returned by local vs. global closed list, not just success/failure.