Planning and Optimization

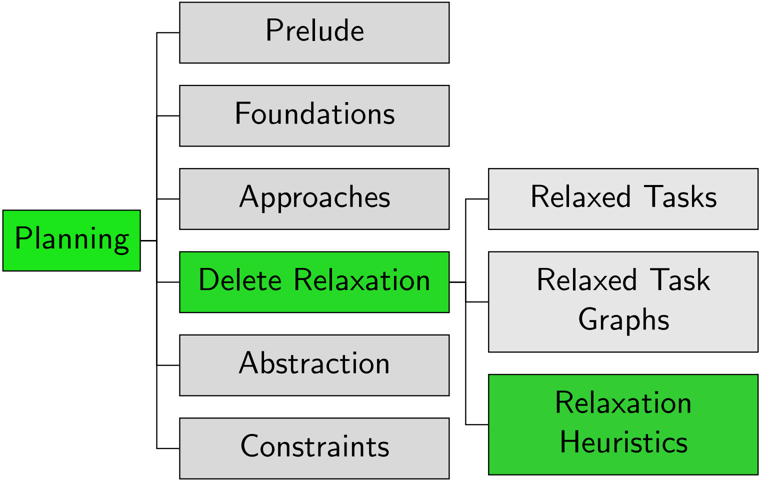
D8. Delete Relaxation: h^{FF} and Comparison of Heuristics

Malte Helmert and Gabriele Röger

Universität Basel

October 29, 2025

Content of the Course



The FF Heuristic

Inaccuracies in h^{max} and h^{add}

- h^{max} is often inaccurate because it undercounts:
  the heuristic estimate only reflects the cost of a critical path, which is often only a small fraction of the overall plan.
- h^{add} is often inaccurate because it overcounts:
  if the same subproblem is reached in many ways, it will be counted many times although it only needs to be solved once.

The FF Heuristic

With best achiever graphs, there is a simple solution to the overcounting of h^{add}: count all effect nodes that h^{add} would count, but only count each of them once.

Definition (FF Heuristic)

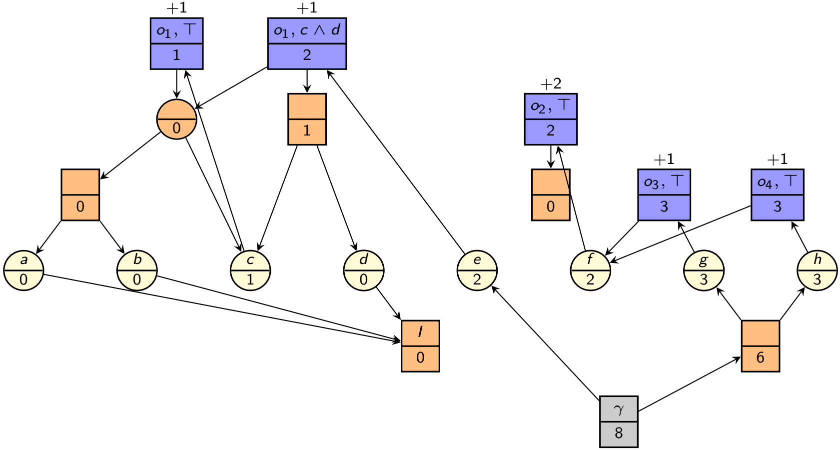
Let $\Pi = \langle V, I, O, \gamma \rangle$ be a propositional planning task in positive normal form. The FF heuristic for a state s of Π, written $h^{\text{FF}}(s)$, is computed as follows:

- Construct the RTG for the task $\langle V, s, O^{+}, \gamma \rangle$
- Construct the best achiever graph G^{add}.
- Compute the set of effect nodes $\{n_{o_1}^{\chi_1}, \dots, n_{o_k}^{\chi_k}\}$ reachable from n_γ in G^{add}.
- Return $h^{\text{FF}}(s) = \sum_{i=1}^{k} cost(o_i)$.

Note: h^{FF} is not well-defined; different tie-breaking policies for best achievers can lead to different heuristic values

Example: FF Heuristic (1)

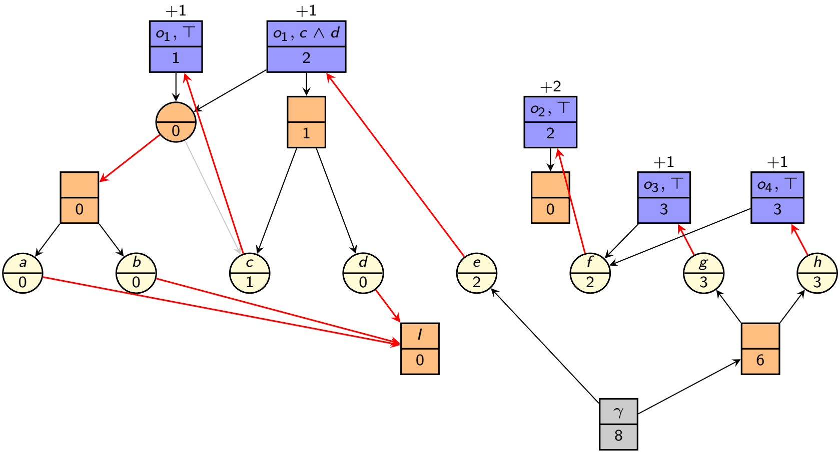
FF heuristic computation



Construct RTG.

Example: FF Heuristic (1)

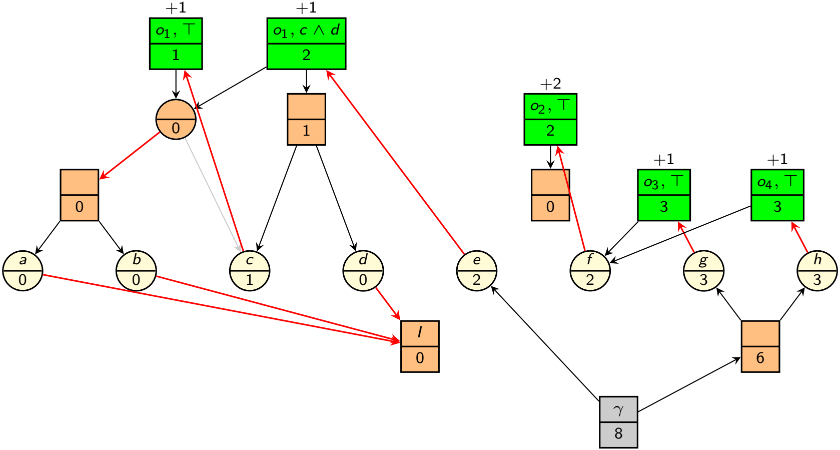
FF heuristic computation



Construct best achiever graph G^{add}.

Example: FF Heuristic (1)

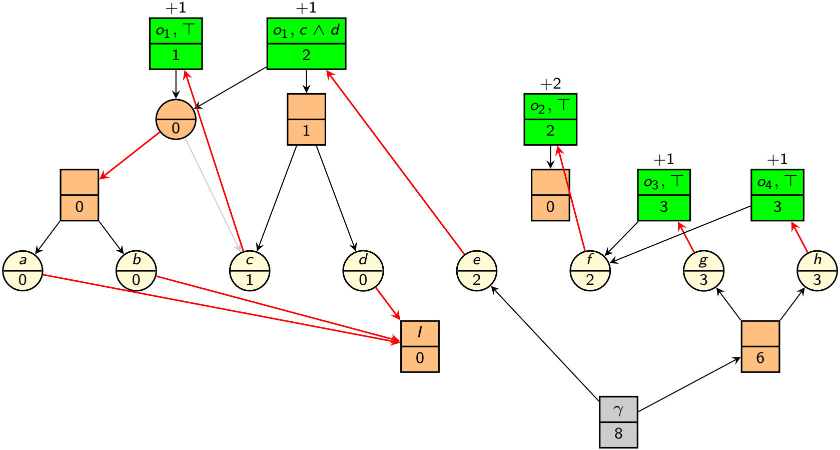
FF heuristic computation



Compute effect nodes reachable from goal node.

Example: FF Heuristic (1)

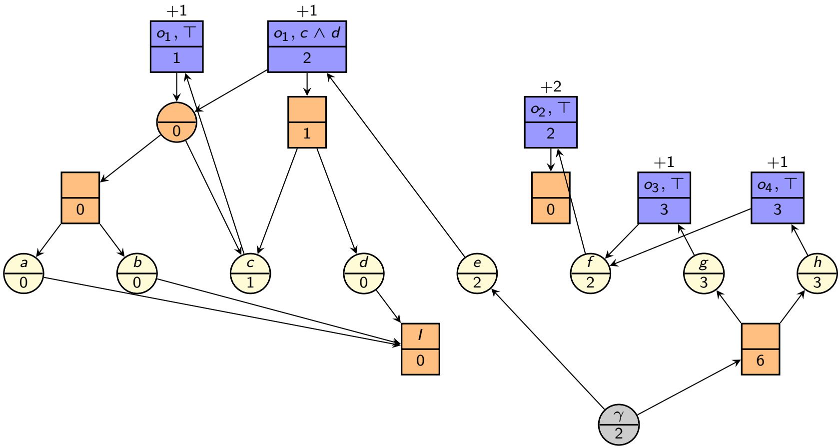
FF heuristic computation



$h^{\text{FF}}(s) = 1 + 1 + 2 + 1 + 1 = 6$

Example: FF Heuristic (2)

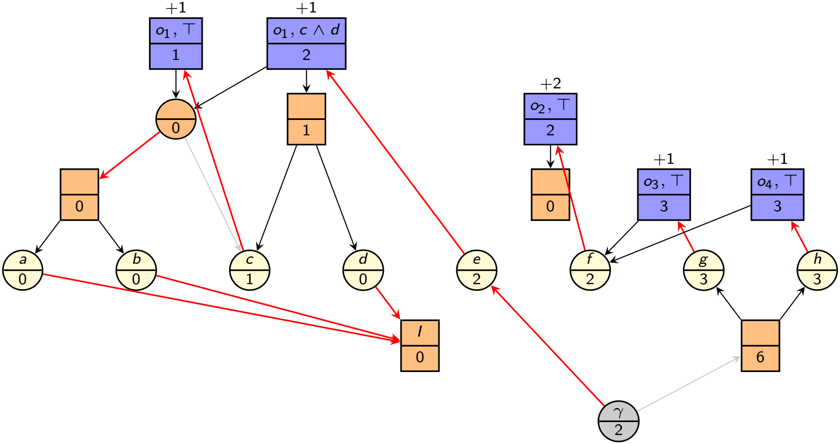
FF heuristic computation; modified goal $e \vee (g \wedge h)$



Construct RTG.

Example: FF Heuristic (2)

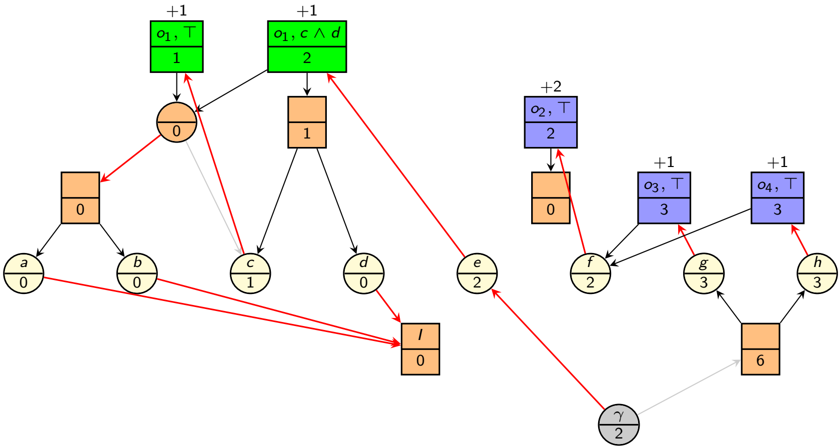
FF heuristic computation; modified goal $e \vee (g \wedge h)$



Construct best achiever graph G^{add}.

Example: FF Heuristic (2)

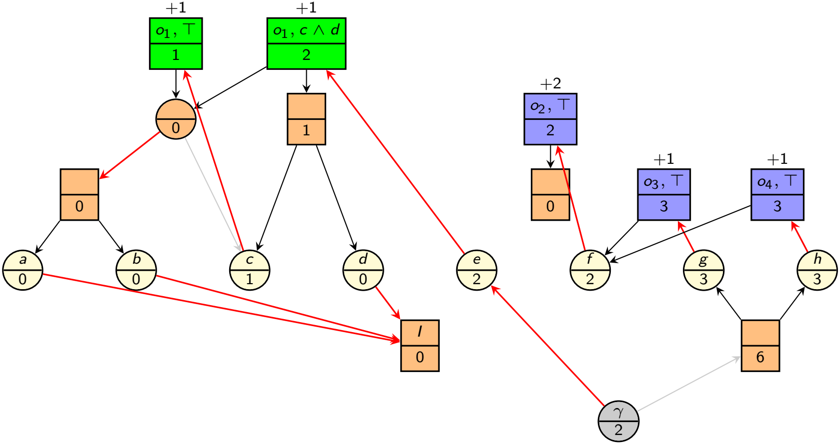
FF heuristic computation; modified goal $e \lor (g \land h)$



Compute effect nodes reachable from goal node.

Example: FF Heuristic (2)

FF heuristic computation; modified goal $e \vee (g \wedge h)$



$h^{\text{FF}}(s) = 1 + 1 = 2$

$h^{\max}$ vs. h^{add} vs. h^{FF} vs. h^{+}

Reminder: Optimal Delete Relaxation Heuristic

Definition (h^+ Heuristic)

Let Π be a propositional planning task in positive normal form, and let s be a state of Π.

The optimal delete relaxation heuristic for s, written $h^+(s)$, is the perfect heuristic value $h^*(s)$ of state s in the delete-relaxed task Π^+.

- Reminder: We proved that $h^+(s)$ is hard to compute. ($\textsc{BCPlanEx}$ is NP-complete for delete-relaxed tasks.)
- The optimal delete relaxation heuristic is often used as a reference point for comparison.

Relationships between Delete Relaxation Heuristics (1)

Theorem

Let Π be a propositional planning task in positive normal form, and let s be a state of Π.

Then:

1. $h^{max}(s) \leq h^{+}(s) \leq h^{FF}(s) \leq h^{add}(s)$
2. $h^{max}(s) = \infty$ *iff* $h^{+}(s) = \infty$ *iff* $h^{FF}(s) = \infty$ *iff* $h^{add}(s) = \infty$
3. h^{max} *and* h^{+} *are admissible and consistent.*
4. h^{FF} *and* h^{add} *are neither admissible nor consistent.*
5. *All four heuristics are safe and goal-aware.*

Relationships between Delete Relaxation Heuristics (2)

Proof Sketch.

for 1:

- To show $h^{\max}(s) \leq h^{+}(s)$, show that critical path costs can be defined for arbitrary relaxed plans and that the critical path cost of a plan is never larger than the cost of the plan.
  Then show that $h^{\max}(s)$ computes the minimal critical path cost over all delete-relaxed plans.
- To show $h^{+}(s) \leq h^{\text{FF}}(s)$, prove that the operators belonging to the effect nodes counted by h^{FF} form a relaxed plan.
  No relaxed plan is cheaper than h^{+} by definition of h^{+}.
- $h^{\text{FF}}(s) \leq h^{\text{add}}(s)$ is obvious from the description of h^{FF}:
  both heuristics count the same operators,
  but h^{add} may count some of them multiple times.

...

Relationships between Delete Relaxation Heuristics (3)

Proof Sketch (continued).

for 2: all heuristics are infinite iff the task has no relaxed solution

for 3: admissibility follows from $h^{\max}(s) \leq h^{+}(s)$
because we already know that h^{+} is admissible;
we omit the argument for consistency

for 4: construct a counterexample to admissibility for h^{FF}

for 5: goal-awareness is easy to show; safety follows from 2.+3. $\square$

Summary

Summary

- The FF heuristic repairs the double-counting of h^{add} and therefore approximates h^{+} more closely.
- The key idea is to mark all effect nodes "used" for the h^{add} value of the goal and count each of them once.
- In general, $h^{\max}(s) \leq h^{+}(s) \leq h^{\text{FF}}(s) \leq h^{\text{add}}(s)$.
- $h^{\max}$ and h^{+} are admissible; h^{FF} and h^{add} are not.

Literature Pointers

(Some) delete-relaxation heuristics in the planning literature:

- additive heuristic h^{add} (Bonet, Loerincs & Geffner, 1997)
- maximum heuristic $h^{\max}$ (Bonet & Geffner, 1999)
- (original) FF heuristic (Hoffmann & Nebel, 2001)
- cost-sharing heuristic h^{cs} (Mirkis & Domshlak, 2007)
- set-additive heuristics h^{sa} (Keyder & Geffner, 2008)
- FF/additive heuristic h^{FF} (Keyder & Geffner, 2008)
- local Steiner tree heuristic h^{lst} (Keyder & Geffner, 2009)

$\rightsquigarrow$ also hybrids such as semi-relaxed heuristics
and delete-relaxation landmark heuristics