04 Delete Relaxation

Introduction to Relaxation

Delete relaxation is a method to relax planning tasks and automatically compute a heuristic function h

- Delete relaxation is a widely used heuristic, highly successful for satisficing planning
- Note that every h function typically yields good performance in only in *some* domains (balacing search reduction versus computational overhead).
  - Therefore we aim for multiple alternative relaxation methods, depending upon domain

Relaxation Types

Four known types of relxation that generate heuristics include

- Critical path heuristics
- Delete relaxation (covered in this Module)
- Abstractions
- Landmarks

Relaxing the World by Ignoring Delete Lists

"What was once true remains true forever."

Relaxed world: (after)

The Delete Relaxation

Definition: The Delete Relaxation

- We denote a STRIPS action a^+, corresponding a, as the *delete relaxed* action, defined by $pre_{a^+} := pre_a$, $add_{a^+} := add_a$, and $del_{a^+} = \emptyset$.

- A set of actions A^+ corresponds to a set of *relaxed actions* $A^+ := \{a^+ \mid a \in A\}$; and for a sequence of actions $\vec{a} = \langle a_1, \ldots, a_n \rangle$ denoted by $\vec{a}^+$, we denote the corresponding sequence of relaxed actions as $\vec{a}^+ := \langle a_1^+, \ldots, a_n^+ \rangle$.

- For STRIPS planning task $\Pi = (F, A, c, I, G)$ we denote the *delete relaxed planning task* as $\Pi^+ := (F, A^+, c, I, G)$.

- The "$^+$" super-script denotes *delete relaxed* states encountered in the relaxed version of the problem, e.g. s^+ is a fact set just like s.
- The "$^+$" symbol is chosen to capture the additive (monotone) aspect of delete relaxed planning problems.

The Relaxed Plan

Definition: The Relaxed Plan

Let $\Pi = (F, A, c, I, G)$ be a STRIPS planning task, and let s be a state.

- An optimal *relaxed plan* for s is an optimal plan for Π_s^+.
- A relaxed plan for I is also called a relaxed plan for Π.

Question: Do you recall what Π_s is?:

- $\Pi_s = (F, A, c, s, G)$

A Relaxed Plan for "TSP" in Australia

Initial state: $\{at(Sy), v(Sy)\}$

- **Apply** $drive(Sy, Br)^+$: $\{at(Br), v(Br), at(Sy), v(Sy)\}$
- **Apply** $drive(Sy, Ad)^+$: $\{at(Ad), v(Ad), at(Br), v(Br), at(Sy), v(Sy)\}$
- **Apply** $drive(Ad, Pe)^+$: $\{at(Pe), v(Pe), at(Ad), v(Ad), at(Br), v(Br), at(Sy), v(Sy)\}$
- **Apply** $drive(Ad, Da)^+$: $\{at(Da), v(Da), at(Pe), v(Pe), at(Ad), v(Ad), at(Br), v(Br), at(Sy), v(Sy)\}$

Definition of State Dominance

Definition: Dominance

Let $\Pi^+ = (F, A^+, c, I, G)$ be a STRIPS planning task, and let $s^+ ands'^+$ be states. We say that state s'^+ *dominates* state s^+ if $s'^+ \supseteq s^+$, meaning every fact true in s^+ is also true in s'^+.

For example, on the previous slide, what state dominates what?

- Each state along the relaxed plan dominates the previous one, because the actions don't delete any facts.

State Dominance Properties

State Dominance Properties

Let $\Pi^+ = (F, A^+, c, I, G)$ be a *delete relaxed* STRIPS planning task, and let s^+ and s'^+ be states such that s'^+ dominates s^+.

i. If s^+ is a goal state, then s'^+ is also a goal state.

ii. If an action sequence $\vec{a}^+$ is applicable in s^+, then it is also applicable in s'^+, and the state resulting from applying $\vec{a}^+$ in s'^+ *dominates* the state resulting from applying $\vec{a}^+$ in s^+.

Proof: (i) is trivial. (ii) by induction over the length n of $\vec{a}^+$. The base case $n = 0$ is trivial. The inductive case $n \to n+1$ follows directly from the induction hypothesis and the definition of $appl(.\,,.\,)$. Note that it is always better to have more facts true.

The Delete Relaxation & Admissibility

Proposition: Action Dominance

Let $\Pi = (F, A, c, I, G)$ be a STRIPS planning task, let s be a state, and let $a \in A$. Then $appl(s, a^+)$ dominates both (i) s, and (ii) $appl(s, a)$.

Proof. Trivial from the definitions of $appl(s, a)$ and a^+.

Proposition: Relaxed plan

Let $\Pi = (F, A, c, I, G)$ be a STRIPS planning task, let s be a state, and let $\vec{a}$ be a plan for Π_s. Then $\vec{a}^+$ is a *relaxed plan* for Π_s^+.

Proof: Prove by induction over the length of $\vec{a}$ that $appl(s, \vec{a}^+)$ dominates $appl(s, \vec{a})$. Base case is trivial, inductive case follows from (ii) above.

The Delete Relaxation & Admissibility (continued)

Proposition: Delete Relaxation is Admissible

Let $\Pi = (F, A, c, I, G)$ be a STRIPS planning task, let s be a state, and let $\vec{a}$ be a plan for Π_s. Then $\vec{a}^+$ is a relaxed plan for s.

Therefore optimal relaxed plans admissibly estimate the cost of optimal plans

- Applying a relaxed action can only ever make more facts true ((i) above).
- That can only be good, i.e., cannot render the task unsolvable (dominance proposition). [

So how do we find a relaxed plan?:

- We keep applying relaxed actions, and only stop if the goal is true.

Greedy Relaxed Planning

Greedy Relaxed Planning for Π_s^+

$s^+ := s; \vec{a}^+ := \langle\rangle$
while $G \not\subseteq s^+$ **do:**
  if $\exists a \in A$ such that $pre_a \subseteq s^+$ and $appl(s^+, a^+) \neq s^+$ **then**
    select one such a
    $s^+ := appl(s^+, a^+); \vec{a}^+ := \vec{a}^+ \circ \langle a^+ \rangle$
  else return "Π_s^+ is unsolvable" **endif**
endwhile
return $\vec{a}^+$

Proposition Greedy relaxed planning is sound, complete, and terminates in time polynomial in the size of Π.

Proof: Soundness: If $\vec{a}^+$ is returned then, by construction, $G \subseteq appl(s, \vec{a}^+)$. Completeness: If "$\Pi_s^+$ is unsolvable" is returned, then no relaxed plan exists for s^+ at that point; since s^+ dominates s, by the dominance proposition this implies that no relaxed plan can exist for s. Termination: Every $a \in A$ can be selected at most once because afterwards $appl(s^+, a^+) = s^+$.

- It is easy to decide whether a relaxed plan exists

Greedy Relaxed Planning to Generate a Heuristic Function?

Using greedy relaxed planning to generate h

- In search state s during forward search, run greedy relaxed planning on Π_s^+.
- Set $h(s)$ to the cost of $\vec{a}^+$, or ∞ if "Π_s^+ is unsolvable" is returned.

Is this heuristic safe?:

- Yes: $h(s) = \infty$ only if no relaxed plan for s exists, which by admissibility of delete relaxation implies that no plan for s exists.

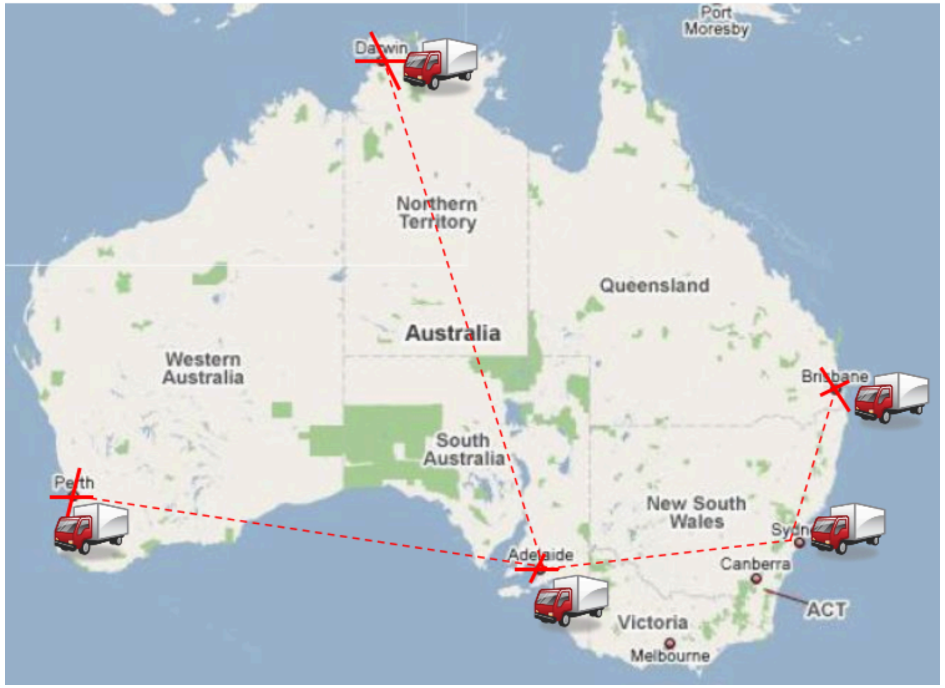
Is this heuristic goal-aware?

- Yes, we'll have $G \subseteq s^+$ right at the start.

Is this heuristic admissible?

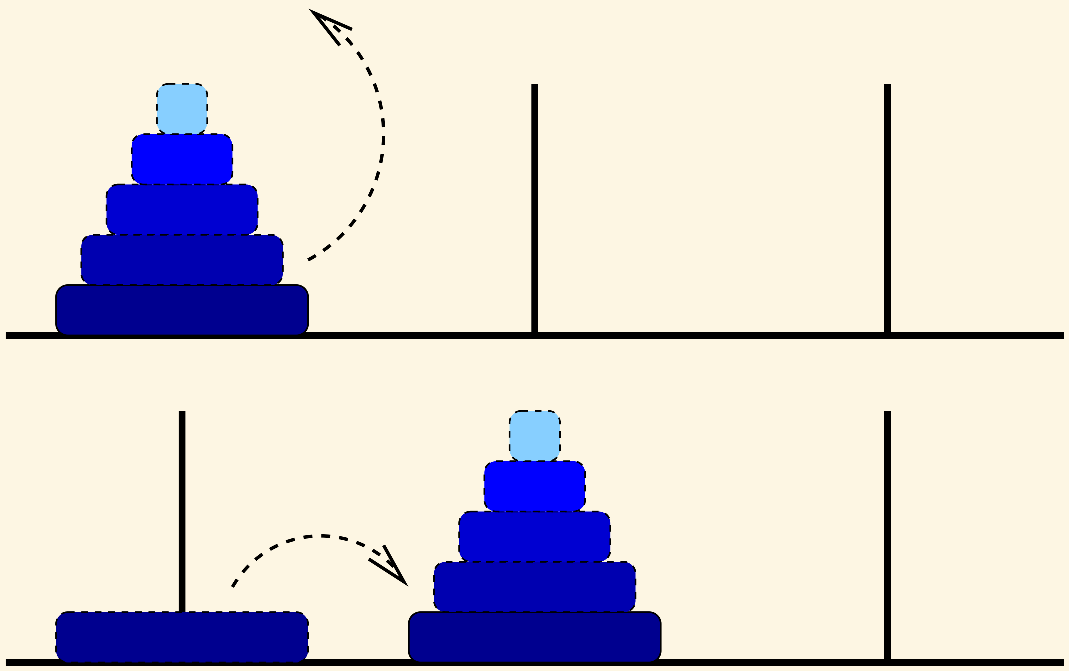
- Would be if relaxed plans were optimal; but clearly aren't. So h isn't consistent either.
- To be informed (to be an accurate estimate of h^*), a heuristic must approximate the needed to reach the goal. Greedy relaxed planning therefore doesn't do this because it may select arbitrary actions that aren't relevant at all.

h^+ in Travelling Saleman Problem (TSP)



$h^+(\text{TSP})$ = Minimum Spanning Tree (every vertex is connected, there are no cycles, and total sum of the edge weights is as small as possible)

h^+ in Towers of Hanoi



$h^+(\text{Hanoi}) = O(n)$, as opposed to problem complexity of $O(2^n)$