Automated Planning

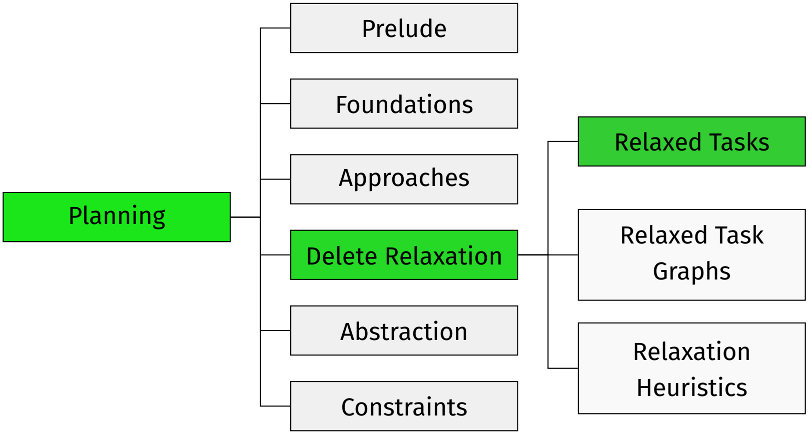
D2. Delete Relaxation: Finding Relaxed Plans

Jendrik Seipp

Linköping University

based on slides from the AI group at the University of Basel

Content of this Course



Greedy Algorithm

On-Set and Dominating States

Definition (On-Set)

The on-set of an interpretation s is the set of propositional variables that are true in s.

$\leadsto$ for states of propositional planning tasks:
states can be viewed as sets of (true) state variables

Definition (Dominate)

An interpretation s' dominates an interpretation s if $on(s) \subseteq on(s')$.

$\leadsto$ all state variables true in s are also true in s'

Motivation

- A general way to come up with heuristics is to solve a simplified version of the real problem.
- delete relaxation: given a task in positive normal form, discard all delete effects
- relaxation lemma: solutions for a state s also work for any dominating state s' (which satisfies a superset of the variables in s)
- monotonicity lemma: $s[\![o]\!]$ dominates s

Greedy Algorithm for Relaxed Planning Tasks

The relaxation and monotonicity lemmas suggest the following algorithm for solving relaxed planning tasks:

Greedy Planning Algorithm for $\langle V, I, O^+, \gamma \rangle$

$s := I$
$\pi^+ := \langle \rangle$
loop forever:
  if $s \models \gamma$:
    return π^+
  else if there is an operator $o^+ \in O^+$ applicable in s
      with $s[\![o^+]\!] \neq s$:
    Append such an operator o^+ to π^+.
    $s := s[\![o^+]\!]$
  else:
    return unsolvable

Correctness of the Greedy Algorithm

The algorithm is sound:

- If it returns a plan, this is indeed a correct solution.
- If it returns "unsolvable", the task is indeed unsolvable
  - Upon termination, there clearly is no relaxed plan from s.
  - By iterated application of the monotonicity lemma, s dominates I.
  - By the relaxation lemma, there is no solution from I.

What about completeness (termination) and runtime?

- Each iteration of the loop adds at least one atom to $on(s)$.
- This guarantees termination after at most $|V|$ iterations.
- Thus, the algorithm can clearly be implemented to run in polynomial time.
  - A good implementation runs in $O(\|\Pi\|)$.

Using the Greedy Algorithm as a Heuristic

We can apply the greedy algorithm within heuristic search for a general (non-relaxed) planning task:

- When evaluating a state s in progression search, solve relaxation of planning task with initial state s.
- When evaluating a subgoal φ in regression search, solve relaxation of planning task with goal φ.
- Set $h(s)$ to the cost of the generated relaxed plan.
  - in general not well-defined: different choices of o^+ in the algorithm lead to different $h(s)$

Is this admissible/safe/goal-aware/consistent?

Using the Greedy Algorithm as a Heuristic

We can apply the greedy algorithm within heuristic search for a general (non-relaxed) planning task:

- When evaluating a state s in progression search, solve relaxation of planning task with initial state s.
- When evaluating a subgoal φ in regression search, solve relaxation of planning task with goal φ.
- Set $h(s)$ to the cost of the generated relaxed plan.
  - in general not well-defined: different choices of o^+ in the algorithm lead to different $h(s)$

Is this admissible/safe/goal-aware/consistent?

It is safe (from the previous slide) and easily seen to be goal-aware. It is not admissible (see next slide) and therefore also not consistent.

Properties of the Greedy Algorithm as a Heuristic

Is this an admissible heuristic?

- Yes if the relaxed plans are optimal (due to the plan preservation corollary).
- However, usually they are not, because the greedy algorithm can make poor choices of which operators to apply.

How hard is it to find optimal relaxed plans?

Optimal Relaxed Plans

Optimal Relaxation Heuristic

Definition (h^+ heuristic)

Let $\Pi = \langle V, I, O, \gamma \rangle$ be a planning task in positive normal form with states S.

The optimal delete relaxation heuristic h^+ for Π
is the function $h : S \longrightarrow \mathbb{R}_0^+ \cup \{\infty\}$
where $h(s)$ is the cost of an optimal relaxed plan for s,
i.e., of an optimal plan for $\Pi_s^+ = \langle V, s, O^+, \gamma \rangle$.

(can analogously define a heuristic for regression)

h^+ is admissible, safe, goal-aware, and consistent.

Complexity of Optimal Relaxed Planning (1)

Theorem (Complexity of Optimal Relaxed Planning)

The BCPLANEX problem restricted to delete-relaxed planning tasks is NP-complete.

Summary

Summary

- Because of their monotonicity property, delete-relaxed tasks can be solved in polynomial time by a greedy algorithm.
- However, the solution quality of this algorithm is poor.
- For an informative heuristic, we would ideally want to find optimal relaxed plans.
- The solution cost of an optimal relaxed plan is the estimate of the h^+ heuristic.
- However, the bounded-cost plan existence problem for relaxed planning tasks is NP-complete.