Artificial Intelligence

Planning 3: Delete Relaxation

Jendrik Seipp

Linköping University

based on slides by Thomas Keller and Malte Helmert (University of Basel)

Questions?

post feedback and ask questions anonymously at

`https://padlet.com/jendrikseipp/tddc17`

Intended Learning Outcomes

- contrast normal STRIPS tasks with “delete-relaxed” STRIPS tasks
- compute $h^{\max}$, h^{add} and h^{FF} for delete-relaxed tasks
- compare the $h^{\max}$, h^{add} and h^{FF} heuristics

Delete Relaxation

Planning Heuristics

General Procedure for Obtaining a Heuristic

Solve a simplified version of the problem.

there are many ideas for domain-independent planning heuristics:

- abstraction $\rightsquigarrow$ yesterday
- delete relaxation $\rightsquigarrow$ now
- landmarks
- critical paths
- network flows
- potential heuristics

Planning Heuristics

Delete Relaxation: Idea

Estimate solution costs by considering a simplified planning task where all negative action effects are ignored.

there are many ideas for domain-independent planning heuristics:

- abstraction $\rightsquigarrow$ yesterday
- delete relaxation $\rightsquigarrow$ now
- landmarks
- critical paths
- network flows
- potential heuristics

Relaxed Planning Tasks: Idea

In STRIPS tasks, good and bad effects are easy to distinguish:

- add effects are always useful
- delete effects are always harmful

Why?

Relaxed Planning Tasks: Idea

In STRIPS tasks, good and bad effects are easy to distinguish:

- add effects are always useful
- delete effects are always harmful

Why? more facts true $\longrightarrow$ more actions applicable

Relaxed Planning Tasks: Idea

In STRIPS tasks, good and bad effects are easy to distinguish:

- add effects are always useful
- delete effects are always harmful

Why? more facts true ⟶ more actions applicable

idea for designing heuristics: ignore all delete effects

Relaxed Planning Tasks

Definition (relaxation of actions)

The relaxation a^+ of STRIPS action a is the action with $pre(a^+) = pre(a)$, $add(a^+) = add(a)$, $cost(a^+) = cost(a)$,
and $del(a^+) = \emptyset$.

Definition (relaxation of planning tasks)

The relaxation Π^+ of a STRIPS planning task $\Pi = \langle V, I, G, A \rangle$
is the task $\Pi^+ := \langle V, I, G, \{a^+ \mid a \in A\} \rangle$.

Definition (relaxation of action sequences)

The relaxation of action sequence $\pi = \langle a_1, \dots, a_n \rangle$
is the action sequence $\pi^+ := \langle a_1^+, \dots, a_n^+ \rangle$.

Relaxed Planning Tasks: Terminology

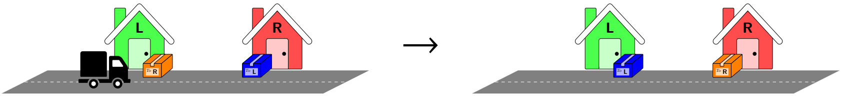
- STRIPS planning tasks without delete effects are called relaxed planning tasks or delete-free planning tasks
- plans for relaxed planning tasks are called relaxed plans
- if Π is a STRIPS planning task and π^+ is a plan for Π^+, then π^+ is called relaxed plan for Π

Relaxed Planning Tasks: Terminology

- STRIPS planning tasks without delete effects are called relaxed planning tasks or delete-free planning tasks
- plans for relaxed planning tasks are called relaxed plans
- if Π is a STRIPS planning task and π^+ is a plan for Π^+, then π^+ is called relaxed plan for Π
- $h^+(\Pi)$ denotes the cost of an optimal plan for Π^+, i.e., of an optimal relaxed plan
- analogously: $h^+(s)$ cost of optimal relaxed plan starting in state s (instead of initial state)
- h^+ is called optimal relaxation heuristic

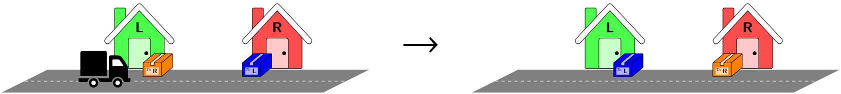
Examples

Example: Logistics



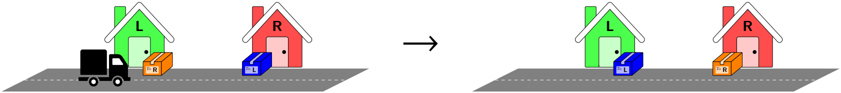
- $V = \{at_{OL}, at_{OR}, at_{BL}, at_{BR}, at_{TL}, at_{TR}, in_{OT}, in_{BT}\}$
- $I = \{at_{OL}, at_{BR}, at_{TL}\}$
- $G = \{at_{OR}, at_{BL}\}$
- $A = \{move_{LR}, move_{RL}, load_{OL}, load_{OR}, load_{BL}, load_{BR}, unload_{OL}, unload_{OR}, unload_{BL}, unload_{BR}\}$
- ...

Example: Logistics



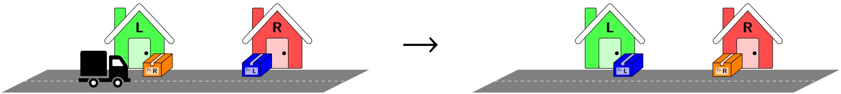
- $pre(move_{LR}) = \{at_{TL}\}, add(move_{LR}) = \{at_{TR}\},$
  $del(move_{LR}) = \{at_{TL}\}, cost(move_{LR}) = 1$
- $pre(load_{OL}) = \{at_{TL}, at_{OL}\}, add(load_{OL}) = \{in_{OT}\},$
  $del(load_{OL}) = \{at_{OL}\}, cost(load_{OL}) = 1$
- $pre(unload_{OL}) = \{at_{TL}, in_{OT}\}, add(unload_{OL}) = \{at_{OL}\},$
  $del(unload_{OL}) = \{in_{OT}\}, cost(unload_{OL}) = 1$
- ...

Example: Logistics



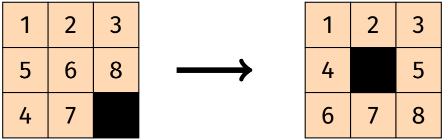
- optimal plan:
  1. $load_{OL}$
  2. $move_{LR}$
  3. $unload_{OR}$
  4. $load_{BR}$
  5. $move_{RL}$
  6. $unload_{BL}$
- optimal relaxed plan: ?
- $h^*(I) = 6$, $h^+(I) = $?

Example: Logistics



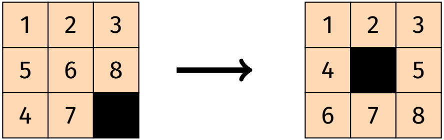
- optimal plan:
  1. $load_{\mathrm{OL}}$
  2. $move_{\mathrm{LR}}$
  3. $unload_{\mathrm{OR}}$
  4. $load_{\mathrm{BR}}$
  5. $move_{\mathrm{RL}}$
  6. $unload_{\mathrm{BL}}$
- optimal relaxed plan: like optimal plan without $move_{\mathrm{RL}}$
- $h^*(I) = 6$, $h^+(I) = 5$

Example: 8-Puzzle



- (original) task:
  - A tile can be moved from cell A to B
    if A and B are adjacent and B is free.
- simplification (basis for Manhattan distance):
  - A tile can be moved from cell A to B
    if A and B are adjacent.
- relaxed task:
  - A tile can be moved from cell A to B if
    A and B are adjacent and B is free.
  - ...where delete effects are ignored
    (in particular: free cells at earlier time remain free)

Example: 8-Puzzle



- actual goal distance: $h^*(s) = 8$
- Manhattan distance: $h^{\mathrm{MD}}(s) = 6$
- optimal delete relaxation: $h^+(s) = 7$

relationship:

h^+ dominates the Manhattan distance in the sliding tile puzzle (i.e., $h^{\mathrm{MD}}(s) \leq h^+(s) \leq h^*(s)$ for all states s)

Exercise

Consider the STRIPS formalization of blocks world and the following task with blocks A, B and C, initial state $I = \{on\text{-}table_A, on_{B,A}, on_{C,B}, clear_C\}$ (left stack in the picture below) and the goal $G = \{on\text{-}table_A, on_{C,A}, on_{B,C}\}$ (right stack in the picture below).

(a) Calculate the perfect heuristic values $h^*(I)$ and $h^*(I')$ for the initial state I and the only successor state I' of I.

(b) Consider the STRIPS heuristic h^S. Calculate the heuristic values $h^S(I)$ and $h^S(I')$.

(c) Calculate $h^+(I)$ and $h^+(I')$.

(d) Compare and discuss the results of exercise parts (a), (b) and (c).

Exercise: Solution

The only successor state of I is $I' = \{on\text{-}table_A, on_{B,A}, on\text{-}table_C\}$.

(a) The following plan is optimal for I: $\langle to\text{-}table_{C,B}, to\text{-}table_{B,A}, from\text{-}table_{C,A}, from\text{-}table_{B,C}\rangle$. Therefore $h^*(I) = 4$. Since the plan starts with the action that reaches I', we have $h^*(I') = 3$.

(b) The goal variables $on_{C,A}$ and $on_{B,C}$ do not hold in I nor in I', so $h^S(I) = h^S(I') = 2$.

(c) To calculate h^+, we inspect the relaxed planning task Π^+. To reach G from I in Π^+, we need 3 actions: $to\text{-}table_{C,B}$, $move_{B,A,C}$ and $from\text{-}table_{C,A}$. Thus $h^+(I) = 3$. Since we already applied $to\text{-}table_{C,B}$ to reach I', we have $h^+(I') = 2$.

(d) The STRIPS heuristic only changes between two states if a goal variable becomes true or false, so $h^S(I) = h^S(I')$. Since the STRIPS heuristic also ignores the actions, it underestimates the effort to reach the goal: $h^S(I) = 2 < h^*(s) = 4$.
In the delete relaxation, C remains clear even when moving B from A to C in the second step. This is not possible in the original task.

Relaxed Solutions: Suboptimal or Optimal?

- for general STRIPS planning tasks, h^+ is an admissible and consistent heuristic

Relaxed Solutions: Suboptimal or Optimal?

- for general STRIPS planning tasks, h^+ is an admissible and consistent heuristic
- Can h^+ be computed efficiently?
  - it is easy to solve delete-free planning tasks suboptimally
  - optimal solution (and hence the computation of h^+) is NP-hard
- in practice, heuristics approximate h^+ from below or above