Basics of AI and Machine Learning

Planning Heuristics

David Speck

Linköping University

Slides modified from Basel AI group, with permission

How to Design Heuristics?

A Simple Planning Heuristic: Goal Count

The STRIPS planner (Fikes & Nilsson, 1971) uses the number of goals not yet satisfied in a STRIPS planning task as heuristic:

$$h(s) := |G \setminus s|.$$

intuition: fewer unsatisfied goals $\rightsquigarrow$ closer to goal state

A Simple Planning Heuristic: Goal Count

The STRIPS planner (Fikes & Nilsson, 1971) uses the number of goals not yet satisfied in a STRIPS planning task as heuristic:

$$h(s) := |G \setminus s|.$$

intuition: fewer unsatisfied goals $\rightsquigarrow$ closer to goal state

drawback of goal count heuristic:

- rather uninformed
- ignores almost the whole task structure:
  The heuristic values do not depend on the actions.

$\rightsquigarrow$ we need better methods to design heuristics

Planning Heuristics

There are different types of general heuristics:

- delete relaxation $\rightsquigarrow$ this chapter
- abstraction $\rightsquigarrow$ not in this course
- landmarks $\rightsquigarrow$ not in this course
- ...

Delete Relaxation: Basic Idea

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

Automated Planning: Overview

Chapter overview: automated planning

- Introduction
- The STRIPS Planning Formalism
- Other Planning Formalisms
- Planning Heuristics
- Alternatives to Heuristic Search

Delete Relaxation

Relaxed Planning Tasks: Idea

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

- Add effects are always useful.
- Delete effects are always harmful.

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 $\Pi^+ := \langle V, I, G, \{a^+ \mid a \in A\} \rangle$.

Relaxed Planning Tasks

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

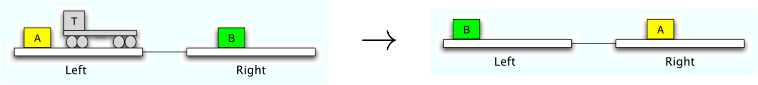
- STRIPS planning tasks without delete effects are called relaxed planning tasks or delete-free planning tasks.
- Plans for relaxed planning tasks are called relaxed plans.
- If Π is a STRIPS planning task and π^+ is a plan for Π^+, then π^+ is called relaxed plan for Π.

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

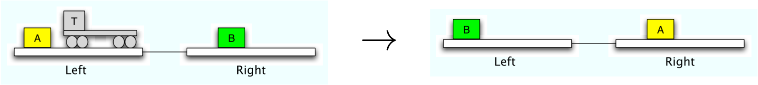
Example: Logistics



Example (Logistics Task)

- variables: $V = \{at_{AL}, at_{AR}, at_{BL}, at_{BR}, at_{TL}, at_{TR}, in_{AT}, in_{BT}\}$
- initial state: $I = \{at_{AL}, at_{BR}, at_{TL}\}$
- goals: $G = \{at_{AR}, at_{BL}\}$
- actions: $\{move_{LR}, move_{RL}, load_{AL}, load_{AR}, load_{BL}, load_{BR}, unload_{AL}, unload_{AR}, unload_{BL}, unload_{BR}\}$
- ...

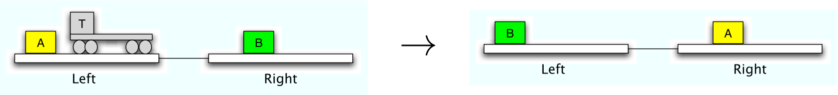
Example: Logistics



Example (Logistics Task)

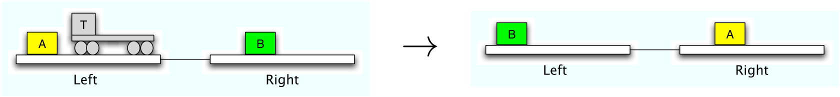
- $pre(move_{\mathsf{LR}}) = \{at_{\mathsf{TL}}\}$, $add(move_{\mathsf{LR}}) = \{at_{\mathsf{TR}}\}$, $del(move_{\mathsf{LR}}) = \{at_{\mathsf{TL}}\}$, $cost(move_{\mathsf{LR}}) = 1$
- $pre(load_{\mathsf{AL}}) = \{at_{\mathsf{TL}}, at_{\mathsf{AL}}\}$, $add(load_{\mathsf{AL}}) = \{in_{\mathsf{AT}}\}$,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

Example: Logistics



- optimal plan:
  1. $load_{\text{AL}}$
  2. $move_{\text{LR}}$
  3. $unload_{\text{AR}}$
  4. $load_{\text{BR}}$
  5. $move_{\text{RL}}$
  6. $unload_{\text{BL}}$
- $h^*(I) = 6$

Example: Logistics



- optimal plan:
  1. $load_{AL}$
  2. $move_{LR}$
  3. $unload_{AR}$
  4. $load_{BR}$
  5. $move_{RL}$
  6. $unload_{BL}$
- $h^*(I) = 6$

- optimal relaxed plan:
  1. $load_{AL}$
  2. $move_{LR}$
  3. $unload_{AR}$
  4. $load_{BR}$
  - ✗ ~~$move_{RL}$~~
  5. $unload_{BL}$
- $h^+(I) = 5$

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. (How?)
  - optimal solution (and hence the computation of h^+) is NP-hard (reduction from SET COVER)
- In practice, heuristics approximate h^+ from below or above.

Summary

Summary

delete relaxation:

- ignore negative effects (delete effects) of actions
- use solution costs of relaxed planning task
  as heuristic for solution costs of the original planning task
- computation of optimal relaxed solution costs h^+ is NP-hard,
  hence usually approximated from below or above