

Grundlagen der Künstlichen Intelligenz

35. Handlungsplanung: Delete-Relaxierung

Malte Helmert

Universität Basel

8. Mai 2015

Grundlagen der Künstlichen Intelligenz

8. Mai 2015 — 35. Handlungsplanung: Delete-Relaxierung

35.1 Woher kommen Heuristiken?

35.2 Delete-Relaxierung

35.3 Beispiele

35.4 Zusammenfassung

35.1 Woher kommen Heuristiken?

Eine einfache Planungsheuristik

STRIPS (Fikes & Nilsson, 1971) verwendete die **Anzahl noch nicht erreichter Ziele** in einer STRIPS-Planungsaufgabe als Heuristik:

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

Intuition: weniger unerreichte Ziele $\rightsquigarrow$ näher an einem Zielzustand

$\rightsquigarrow$ **STRIPS-Heuristik** (Eigenschaften?)

Kritik an der STRIPS-Heuristik

Welche Probleme hat die STRIPS-Heuristik?

- ▶ recht **uninformativ**:
wenn im Zustand s nicht sofort ein Ziel erreicht
(oder verloren) werden kann, haben alle Nachfolger von s
denselben Heuristikwert.
- ▶ ignoriert fast die gesamte **Problemstruktur**:
Der Heuristikwert hängt überhaupt nicht
von den Aktionen ab!

↪ wir brauchen eine bessere allgemeine Methode,
um Heuristiken zu entwickeln

Planungsheuristiken

Wir besprechen **drei Kernideen** für allgemeine Heuristiken:

- ▶ **Delete-Relaxierung** ↪ dieses und nächstes Kapitel
- ▶ **Abstraktion** ↪ später
- ▶ **Landmarken** ↪ später

Grundidee Delete-Relaxierung

Schätze Lösungskosten durch Betrachten
einer **vereinfachten Planungsaufgabe** ab, bei der
unangenehme Aktionseffekte ignoriert werden.

Handlungsplanung: Überblick

Kapitelüberblick:

- ▶ 33. Einführung
- ▶ 34. Planungsformalismen
- ▶ 35.–36. Planungsheuristiken: Delete-Relaxierung
 - ▶ **35. Delete-Relaxierung**
 - ▶ 36. Delete-Relaxierungs-Heuristiken
- ▶ 37.–38. Planungsheuristiken: Abstraktion
- ▶ 39.–40. Planungsheuristiken: Landmarken

35.2 Delete-Relaxierung

Relaxierte Planungsaufgaben: Idee

In STRIPS-Planungsaufgaben sind gute und schlechte Effekte leicht zu unterscheiden:

- ▶ Add-Effekte sind immer **nützlich**.
- ▶ Delete-Effekte sind immer **schädlich**.

Warum?

Idee für die Heuristik: **Ignoriere alle Delete-Effekte**.

Relaxierte Planungsaufgaben

Definition (Relaxierung von Aktionen)

Die **Relaxierung** a^+ einer STRIPS-Aktion a ist die Aktion mit $pre(a^+) = pre(a)$, $add(a^+) = add(a)$, $cost(a^+) = cost(a)$ und $del(a^+) = \emptyset$.

Definition (Relaxierung von Planungsaufgaben)

Die **Relaxierung** Π^+ einer STRIPS-Planungsaufgabe $\Pi = \langle V, I, G, A \rangle$ ist die Planungsaufgabe $\Pi^+ := \langle V, I, G, \{a^+ \mid a \in A\} \rangle$.

Definition (Relaxierung von Aktionsfolgen)

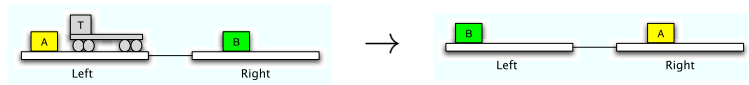
Die **Relaxierung** einer Aktionsfolge $\pi = \langle a_1, \dots, a_n \rangle$ ist die Aktionsfolge $\pi^+ := \langle a_1^+, \dots, a_n^+ \rangle$.

Relaxierte Planungsaufgaben: Terminologie

- ▶ STRIPS-Planungsaufgaben ohne Delete-Effekte heißen **relaxierte Planungsaufgaben** oder **delete-freie Planungsaufgaben**.
- ▶ Pläne für relaxierte Planungsaufgaben heißen **relaxierte Pläne**.
- ▶ Wenn Π eine STRIPS-Planungsaufgabe ist und π^+ ein Plan für Π^+ ist, dann heisst π^+ **relaxierter Plan für Π** .
- ▶ Mit $h^+(\Pi)$ bezeichnen wir die Kosten eines **optimalen Plans** für Π^+ , also eines **optimalen relaxierten Plans**.
- ▶ analog: $h^+(s)$ Kosten für optimalen relaxierten Plan vom Zustand s aus (statt vom Anfangszustand)
- ▶ h^+ heisst **optimale Relaxierungsheuristik**.

35.3 Beispiele

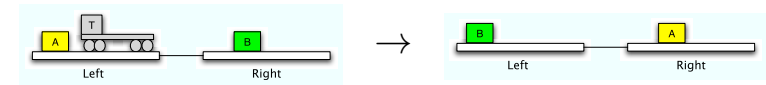
Beispiel: Logistik



Beispiel (Logistikproblem)

- **Variablen:** $V = \{at_{AL}, at_{AR}, at_{BL}, at_{BR}, at_{TL}, at_{TR}, in_{AT}, in_{BT}\}$
- **Anfangszustand:** $I = \{at_{AL}, at_{BR}, at_{TL}\}$
- **Ziele:** $G = \{at_{AR}, at_{BL}\}$
- **Aktionen:** $\{move_{LR}, move_{RL}, load_{AL}, load_{AR}, load_{BL}, load_{BR}, unload_{AL}, unload_{AR}, unload_{BL}, unload_{BR}\}$
- ...

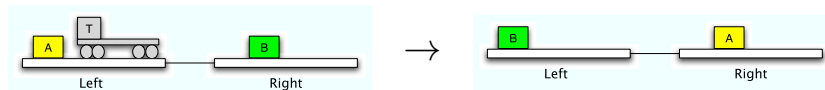
Beispiel: Logistik



Beispiel (Logistikproblem)

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

Beispiel: Logistik



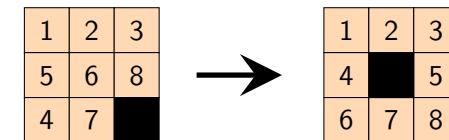
► optimaler Plan:

- 1 $load_{AL}$
- 2 $move_{LR}$
- 3 $unload_{AR}$
- 4 $load_{BR}$
- 5 $move_{RL}$
- 6 $unload_{BL}$

► optimaler relaxierter Plan: ?

- $h^*(I) = 6$, $h^+(I) = ?$

Beispiel: 8-Puzzle



► (Unvereinfachtes) Problem:

- Eine Kachel kann von Feld A in Feld B bewegt werden, wenn A und B angrenzend sind und B frei ist.

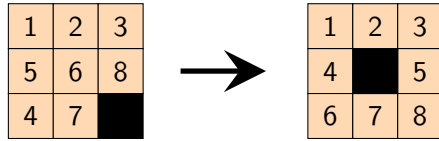
► Vereinfachung, die der Manhattan-Distanz zugrunde liegt:

- Eine Kachel kann von Feld A in Feld B bewegt werden, wenn A und B angrenzend sind.

► relaxiertes Problem:

- Eine Kachel kann von Feld A in Feld B bewegt werden, wenn A und B angrenzend sind und B frei ist.
- ... wobei Delete-Effekte ignoriert werden (insbesondere: Felder, die jemals frei waren, bleiben frei)

Beispiel: 8-Puzzle



- ▶ tatsächlicher Zielabstand: $h^*(s) = 8$
- ▶ Manhattan-Distanz: $h^{MD}(s) = 6$
- ▶ optimale Delete-Relaxierung: $h^+(s) = 7$

allgemein:

h^+ dominiert Manhattan-Distanz für das Schiebe-Puzzle
(d. h. $h^{MD}(s) \leq h^+(s) \leq h^*(s)$ für alle Zustände s)

Relaxierte Lösungen: suboptimal oder optimal?

- ▶ Für allgemeine STRIPS-Planungsaufgaben ist h^+ eine **zulässige und konsistente Heuristik**.
- ▶ Wie berechnen wir h^+ effizient?
 - ▶ Es ist **einfach**, delete-freie Planungsaufgaben **suboptimal** zu lösen. (Wie?)
 - ▶ Optimale Lösung (und damit Berechnung von h^+) ist aber **NP-hart** (einfache Reduktion von SET COVER).
- ▶ Praktische Heuristiken nähern h^+ von unten oder oben an.

35.4 Zusammenfassung

Zusammenfassung

Delete-Relaxierung:

- ▶ ignoriere **negative Effekte** (Delete-Effekte) von Aktionen
- ▶ verwende **Lösungskosten für die relaxierte Planungsaufgabe** als **Heuristik** für Lösungskosten der eigentlichen Planungsaufgabe
- ▶ optimale relaxierte Lösungskosten h^+ NP-hart zu berechnen, daher normalerweise **Approximation** von oben oder unten