

Was bisher geschah

- ▶ Daten, Information, Wissen
- ▶ Wissensrepräsentation und -verarbeitung

Wissensrepräsentation: Beschreibung von

Wissen: Zustandsübergangssystem:

gerichteter Graph $G = (V, E)$ mit

- ▶ Knotenmarkierungen $l_V : V \rightarrow L_V$ mit L_V : Eigenschaften der Zustände
- ▶ Startzustand $s \in V$
- ▶ Eigenschaften der Zielzustände (z.B. Einschränkung der Variablenwerte)
- ▶ Kantenmarkierungen $l_E : V \rightarrow L_E$ mit L_E : mögliche / zulässige Aktionen (Übergänge)

Lösung: zulässiger Weg (Zustandsfolge $p \in V^*$) vom Start- zu einem Zielzustand

Wissensverarbeitung: Pfadsuche im Graphen

- ▶ uninformierte (blinde) Suchverfahren: Tiefen-, Breitensuche
Schrittweise Vertiefung
- ▶ Gleiche-Kosten-Suche

Heuristische Suche – Motivation

Heuristik: Effizienzsteigerung durch Zusatzinformationen
(z.B. Erfahrungswerte)

Anwendung bei

- ▶ Aufgaben mit mehreren Lösungen (z.B. Wege in Graphen)
- ▶ unterschiedliche Qualität der Lösungen
(z.B. Länge des Weges)
- ▶ Suche nach **optimalen** Lösungen (z.B. kürzester Weg)
- ▶ falls vollständige Suche zu aufwendig

Ziele:

- ▶ Wahl einer geeigneten Such-Reihenfolge, unter welcher gute Lösungen zuerst gefunden werden
- ▶ Verwerfen von Knoten, die wahrscheinlich nicht zu einer Lösung führen
(beabsichtigte Verletzung der Fairness-Eigenschaft)

Schätzfunktionen

Ziel: sinnvolle Auswahl der in jedem Schritt zu expandierenden Knoten unter Verwendung von Zusatzinformationen

Schätzfunktion (heuristische Funktion) $h : V \rightarrow \mathbb{R}_{\geq 0} \cup \{\infty\}$
(oder in eine andere geordnete Menge)
Schätzung der erwartete Restkosten vom Knoten u
bis zum Ziel

repräsentiert die Zusatzinformation

Eigenschaften von Heuristiken

Schätzfunktion $h : V \rightarrow \mathbb{R}_{\geq 0} \cup \{\infty\}$ heißt

perfekt (Schätzfunktion $H(u)$), gdw. $\forall u \in V : H(u) =$
genau die Kosten einer optimalen Lösung durch u
($H(u) = \infty$, falls keine Lösung über u existiert)

zielerkennend gdw. für jeden Lösungsknoten $u \in V$ gilt $h(u) = 0$

sicher gdw. für jeden Knoten $u \in V$, aus dem kein
Lösungsknoten erreichbar ist, gilt $h(u) = \infty$
d.h. $\forall u : (h(u) = \infty \rightarrow H(u) = \infty)$

konsistent gdw. für jeden Knoten $u \in V$ und alle Folgeknoten v
von u gilt $h(u) \leq w(u, v) + h(v)$
($w(u, v)$ Kosten des Übergangs von u nach v)

nicht-überschätzend gdw. für jeden Knoten $u \in V$ gilt
 $h(u) \leq H(u)$

Aus nicht-überschätzend folgt sicher und zielerkennend. (ÜA)

Aus zielerkennend und konsistent folgt nicht-überschätzend. (ÜA)

Besten-Suche

(best-first-search)

Allgemeines Suchverfahren mit Bewertungsfunktion

$$f : V \rightarrow \mathbb{R}_{\geq 0} \cup \{\infty\}$$

mit folgender Strategie zur Auswahl der in jedem Schritt zu expandierenden Knoten:

- ▶ Knoten werden aufsteigend nach Bewertung $f(u)$ expandiert,
- ▶ Expansion des Knotens u mit dem geringsten Wert $f(u)$ zuerst
- ▶ Verwaltung von L_a als priority queue

Beispiel: Suche eines kürzesten Weges zwischen Orten A und B

- ▶ Bewertungsfunktion $f(u)$: bisherige Kosten bis zum Ort u (ohne Schätzfunktion, uniforme Kostensuche, Dijkstra)
- ▶ Bewertungsfunktion $f(u)$:
Luftlinienentfernung des Ortes u von B (nur Schätzfunktion)

Besten-Suche – Eigenschaften

zwei Methoden:

1. Knoten mit großen Werten **möglichst spät** expandieren
 2. Knoten mit großen Werten **nicht** expandieren
-
- ▶ Bestensuche mit einer beliebigen Bewertungsfunktion ist nicht immer optimal.
 - ▶ Bestensuche nach Methode 1 (fair) ist vollständig.
 - ▶ Bestensuche nach Methode 2 ist nicht immer vollständig.

Greedy-Suche (kleinste Restkosten)

Idee: Suche zuerst in Teilbäumen der noch nicht besuchten Knoten mit den geringsten (geschätzten) noch aufzuwendenden Kosten

Heuristische Funktion $h : V \rightarrow \mathbb{R}_{\geq 0} \cup \{\infty\}$

$h(v)$ ist Abschätzung des von Knoten v aus den **noch notwendigen** Kosten zum Erreichen eines Zielzustandes

Greedy-Suche:

Besten-Suche mit Bewertungsfunktion $f : V \rightarrow \mathbb{R}_{\geq 0} \cup \{\infty\}$, wobei für jeden Knoten $v \in V$ gilt

$$f(v) = h(v)$$

Eigenschaften der Greedy-Suche:

- ▶ optimal?
- ▶ vollständig?

Beispiel Schiebefax

- ▶ Zustände $u \in \{0, \dots, 8\}^{3 \times 3}$, 3×3 -Matrix mit Einträgen $\{0, \dots, 8\}$ (jede Zahl genau einmal, 0 leeres Feld)
- ▶ Zulässige Züge: Verschieben des leeren Feldes auf ein Nachbarfeld d. h. Vertauschen von 0 und einem Wert in einem Nachbarfeld (gleicher Zeilen- oder Spaltenindex)
- ▶ Zielkonfiguration

1	2	3
8		4
7	6	5

- ▶ Aufgabeninstanz: gegebene Ausgangskonfiguration (Matrix), z.B.

8		3
2	1	4
7	6	5

- ▶ Lösung: Folge von zulässigen Zügen (Bewegung der Lücke 0) von der Ausgangs- zur Zielkonfiguration
- ▶ Bewertung der Lösung: Anzahl der Züge (Länge der Lösungsfolge)

Schiebefax – Heuristische Funktionen

Heuristische Funktionen $h_i : \{0, \dots, 8\}^{3 \times 3} \rightarrow \mathbb{N}$ mit

- h_1 Anzahl der Zahlen, die sich nicht an ihrer Zielposition befinden
- h_2 weitester Abstand einer Zahl zu seiner Zielposition
- h_3 Summe der Manhattan-Abstände jeder Zahl zu seiner Zielposition

Tafel: Bestensuche mit Bewertungsfunktionen $f(u) = h_i(u)$

Qualität der Schätzfunktionen:

- ▶ gute Trennung verschiedener Zustände
- ▶ fair: zu jedem $n \geq 0$ existieren nur endlich viele $u \in V$ mit $h(u) \leq n$

Bisherige Kosten

Kostenfunktion $k : V \rightarrow \mathbb{R}_{\geq 0}$

$k(u)$ Kosten des besten (bisher bekannten) Pfades vom Startzustand zum Zustand u

Kostenfunktion $k : V \rightarrow \mathbb{R}_{\geq 0}$ heißt

streng monoton wachsend , falls für alle Knoten u und alle Nachfolger v von u gilt $k(u) < k(v)$

Beispiele für Kostenfunktionen:

- ▶ Tiefe des Knotens im Suchbaum,
- ▶ maximale Entfernung vom Startknoten

A*-Suche (kleinste Gesamtkosten)

Idee: Suche zuerst in Teilbäumen der noch nicht besuchten Knoten mit dem **geringsten Wert der Schätzfunktion** (Summe von bisherigen und geschätzten zukünftigen Kosten)

Funktionen

- ▶ $k : V \rightarrow \mathbb{R}_{\geq 0}$ – geringste bisher bekannte Kosten von einem Startzustand zu v
- ▶ $h : V \rightarrow \mathbb{R}_{\geq 0}$ – geschätzte Kosten von v zu einem Endzustand Lösung

A*-Suche:

Besten-Suche mit Schätzfunktion $f : V \rightarrow \mathbb{R}_{\geq 0}$, wobei für jeden Knoten $v \in V$ gilt

$$f(v) = k(v) + h(v)$$

Eigenschaften der A*-Suche:

- ▶ vollständig?
- ▶ optimal?

Anwendungen

Planungsprobleme und kombinatorische Suchprobleme, z.B.

- ▶ Routenplanung
- ▶ TSP
- ▶ Verlegen von Leitungen
- ▶ Schaltkreis-Layout
- ▶ Scheduling
- ▶ Produktionsplanung
- ▶ Navigation (z.B. autonomer Fahrzeuge)