Grundlagen der Künstlichen Intelligenz

7. Suchalgorithmen: Eigenschaften von A*

Malte Helmert

Universität Basel

25. März 2013

7.1 Einleitung

Suchprobleme: Überblick

Kapitelüberblick klassische Suchprobleme:

- Formalisierung von Suchproblemen (⇝ Kapitel 3)
- blinde Suchverfahren (⇝ Kapitel 4)
- Heuristiken (⇝ Kapitel 5)
- Bestensuche (⇝ Kapitel 6)
- Eigenschaften von A* (⇝ dieses Kapitel)
- Lokale Suche (⇝ Kapitel 8)

Eigenschaften von A*

- wichtigster Vorzug von A* gegenüber gieriger Suche: optimal unter geeigneten Forderungen an die Heuristik
- sehr wichtiges Ergebnis!
  $\rightsquigarrow$ wir schauen uns die Gründe dafür im Detail an
- dafür zunächst zwei Lemmas

7.2 Monotonie-Lemma

Eigenschaften von A*: Monotonie

Lemma (Monotonie von A* mit konsistenter Heuristiken)

Betrachte A mit einer konsistenten Heuristik.*
Dann gilt:

1. *Wenn n' ein Kindknoten von n ist, dann $f(n') \geq f(n)$.*
2. *Die f-Werte entlang aller von A* betrachteten Pfade steigen monoton.*
3. *Die f-Werte der Folge der von A* expandierten Knoten steigen monoton.*

Eigenschaften von A*: Monotonie (Fortsetzung)

Beweis.

zu 1.:
Sei n' Kindknoten von n via Aktion a.
Sei $s = n$.state, $s' = n'$.state.

- Definition von f: $f(n) = g(n) + h(s)$, $f(n') = g(n') + h(s')$
- Definition von g: $g(n') = g(n) + cost(a)$
- Konsistenz von h: $h(s) \leq cost(a) + h(s')$

$\rightsquigarrow$ $f(n) = g(n) + h(s) \leq g(n) + cost(a) + h(s')$
$= g(n') + h(s') = f(n')$

zu 2.: folgt sofort aus 1.

...

Eigenschaften von A*: Monotonie (Fortsetzung)

Beweis (Fortsetzung).

zu 3.:

- Sei f_b der minimale f-Wert in *open* zu Beginn einer Schleifeniteration. Sei n der entnommene Knoten mit $f(n) = f_b$.
- zu zeigen: am Ende dieser Iteration ist der minimale f-Wert in *open* mindestens f_b.
- Wir müssen die Operationen betrachten, die *open* modifizieren: *open*.pop-min und *open*.insert.
- Durch *open*.pop-min kann sich der minimale f-Wert in *open* nicht reduzieren (höchstens erhöhen).
- Die mit *open*.insert hinzugefügten Knoten n' sind Kinder von n und erfüllen daher $f(n') \geq f(n) = f_b$ nach Teil 1.

□

7.3 Optimale-Pfade-Lemma

Optimale Pfade

Wir führen einige Begriffe im Zusammenhang mit optimalen Pfaden vom Anfangszustand s_0 zu Zuständen s ein:

- $g^*(s)$: Kosten optimaler Pfad von Anfangszustand zu s
- $f_h^*(s) = g^*(s) + h(s)$

Anmerkungen:

- f_h^* und g^* über Zustände, nicht Knoten definiert, anders als f und g (Warum?)
- $f_h^*(n.\text{state}) \leq f(n)$ für alle von Suchalgorithmen erzeugbaren Knoten n (Warum?)

Eigenschaften von A*: optimale Pfade

Lemma (Optimale Pfade für A* mit konsistenter Heuristik)

Sei n ein von A mit konsistenter Heuristik expandierter Knoten, und sei $s = n.\text{state}$. Dann gilt:*

1. $g(n) = g^*(s)$
2. $f(n) = f_h^*(s)$

In Worten: immer wenn A* mit konsistenter Heuristik den Knoten n expandiert, wurde ein optimaler Pfad von s_0 zum Zustand von n gefunden

Eigenschaften von A*: optimale Pfade (Fortsetzung)

Beweis.

Induktion über die Zahl der expandierten Zustände

Induktionsanfang: Als erstes wird der Wurzelknoten n_0 für Anfangszustand s_0 expandiert.
Es gilt: $g(n_0) = 0 = g^*(s_0)$ und $f(n_0) = 0 + h(s_0) = f_h^*(s_0)$

Induktionsschritt:

zu 2.:
Betrachte Situation unmittelbar bevor n aus *open* entfernt wird.
Wir können annehmen, dass n kein Duplikat ist ($s \notin closed$), denn sonst wird n nicht expandiert.

- Sei $s_0, s_1, \ldots, s_k$ mit $s_k = s$ ein optimaler Pfad von s_0 zu s.
- Wähle j als die grösste Zahl mit $s_j \in closed$ und $s_{j+1} \notin closed$.

...

Eigenschaften von A*: optimale Pfade (Fortsetzung)

Beweis (Fortsetzung).

- So ein j existiert immer, da:
  - $s_0 \in closed$ (Wurzelknoten wurde schon expandiert)
  - $s_k \notin closed$ (sonst wäre n ein Duplikat)
- Da $s_j \in closed$, wurde ein Knoten n_j mit $n_j.\text{state} = s_j$ expandiert.
- Nach Induktionsvoraussetzung gilt $g(n_j) = g^*(s_j)$.
- Wir dürfen daher annehmen, dass die Zustände $s_0, \ldots, s_j$ gerade so gewählt werden, dass sie den Vorfahren von n_j entsprechen, ohne dass der Pfad $s_0, \ldots, s_k$ dadurch suboptimal wird.
- Seien $n_0, \ldots, n_k$ die zum Pfad $\langle s_0, \ldots, s_k \rangle$ gehörigen Knoten (die nicht unbedingt alle von A* erzeugt wurden). ...

Eigenschaften von A*: optimale Pfade (Fortsetzung)

Beweis (Fortsetzung).

- Aus $s_j \in closed$ und $s_{j+1} \notin closed$ folgt $n_{j+1} \in open$:
  - Bei Expansion des Knotens n_j wurde n_{j+1} in *open* eingefügt.
  - Wenn n_{j+1} aus *open* entnommen worden wäre, müsste $s_{j+1} \in closed$ gelten. (Widerspruch!)
- Unmittelbar vor Expansion von n liegen somit n_{j+1} und n beide in *open* und n wird gewählt.
  Daraus folgt $f(n) \leq f(n_{j+1})$.
- Wegen Monotonie von f entlang Pfaden:
  $f(n_{j+1}) \leq f(n_{j+2}) \leq \cdots \leq f(n_k)$
- Wegen Optimalität des Pfades $s_0, \ldots, s_k$:
  $f(n_k) = g(n_k) + h(s_k) = g^*(s_k) + h(s_k) = f_h^*(s_k) = f_h^*(s)$
- Zusammen: $f(n) \leq f_h^*(s)$ und somit $f(n) = f_h^*(s)$
  (sonst Widerspruch zur Definition von f_h^*) ...

Eigenschaften von A*: optimale Pfade (Fortsetzung)

Beweis (Fortsetzung).

zu 1.:

- aus 2.: $f(n) = f_h^*(s)$
- Definition: $g(n) + h(s) = g^*(s) + h(s)$
- Subtraktion von $h(s)$: $g(n) = g^*(s)$

Im letzten Schritt verwenden wir, dass $h(s)$ endlich sein muss, wenn n expandiert wird. □

Zwischenbetrachtung: Optimalität von A*?

- Ziel unserer Untersuchungen war, die Optimalität von A* unter geeigneten Bedingungen zu beweisen.
- Reichen unsere bisherigen Ergebnisse dafür aus?
- Fast! Unser voriges Lemma garantiert:
  Wenn A* einen Zielknoten aus *open* auswählt, dann wurde ein optimaler Pfad zum zugehörigen Zielzustand gefunden.
- Aber vielleicht kann es billigere Pfade zu anderen Zielzuständen geben?
- Ohne weitere Forderungen ist das in der Tat möglich!

7.4 Optimalität von A*

Optimalität von A* ohne Reopening

Satz (Optimalität von A* ohne Reopening)

A* *ohne Reopening ist optimal, wenn die verwendete Heuristik konsistent und zielerkennend ist.*

Beweis.

Wir modifizieren (als Gedankenexperiment) A* so,
dass nach Erreichen eines Zielzustandes nicht abgebrochen wird.

$\rightsquigarrow$ Wir expandieren weiter, bis *open* leer wird

$\rightsquigarrow$ Alle erreichbaren Zustände werden erreicht. ...

Optimalität von A* ohne Reopening (Fortsetzung)

Beweis (Fortsetzung).

Sei n der erste gefundene Zielknoten, s der zugehörige Zielzustand.

- Optimale-Pfade-Lemma: $f(n) = f_h^*(s) = g^*(s) + h(s)$
- h ist zielerkennend: $h(s) = 0$
- zusammen: $f(n) = g^*(s)$

Sei n' ein später erreichter Zielknoten mit Zielzustand s'.

- analog: $f(n') = g^*(s')$
- Monotonie-Lemma (Teil 3): $f(n) \leq f(n')$
- zusammen: $g^*(s) \leq g^*(s')$

Also sind die optimalen Pfadkosten für alle später erreichten Zielzustände mindestens so hoch wie für s.

$\rightsquigarrow$ erste gefundene Lösung ist optimal

$\rightsquigarrow$ Fortsetzung der Suche ist unnötig $\square$

Diskussion: Wofür Konsistenz?

- Wir haben Konsistenz und Zielerkennung gefordert.
- Wir wissen: aus diesen Eigenschaften folgt Zulässigkeit.
- Reicht auch Zulässigkeit allein, damit A* optimal ist?
- Im Allgemeinen nein: zulässige, aber inkonsistente Heuristiken können das Optimale-Pfade-Lemma verletzen, was zu suboptimalen Lösungen führen kann.
- Abhilfe: Reopening

Optimalität von A* mit Reopening

Satz (Optimalität von A* mit Reopening)

A* *mit Reopening ist optimal, wenn die verwendete Heuristik zulässig ist.*

- Wir skizzieren den Beweis nur.
- Ein komplett formaler Beweis wäre etwas aufwändiger, aber insgesamt leichter zu führen als der vorige Beweis zu A* ohne Reopening.

Optimalität von A* mit Reopening (Fortsetzung)

Beweisskizze.

- Seien c^* die Kosten einer optimalen Lösung.
  Seien S^* die Zustände, die Teil einer optimalen Lösung sind.
  Seien $\tilde{S}$ die Zustände mit $f_h^*(s) \leq c^*$.
- Wegen Zulässigkeit: $S^* \subseteq \tilde{S}$.
- Wir zeigen: Zustände $s \notin \tilde{S}$ können erst expandiert werden, nachdem alle Zustände in S^* expandiert wurden.
- Dabei werden (evtl. über Reopening) alle Transitionen innerhalb S^* berücksichtigt und somit alle Pfade gefunden, die ausschliesslich aus Zuständen in S^* bestehen.

Optimalität von A* mit Reopening (Fortsetzung)

Beweisskizze (Fortsetzung).

- Optimale Lösungen sind solche Pfade.
- Daher finden wir eine Lösung, bevor wir $\tilde{S}$ verlassen, und es folgt, dass $\tilde{S}$ nie verlassen wird.
- Es folgt $f_h^*(n) \leq c^*$ für gefundenen Lösungsknoten

$\rightsquigarrow$ Optimalität

□

7.5 Vollständigkeit

Vollständigkeit von A*

Vollständigkeit von A*:

- A* ist für sichere Heuristiken vollständig, da dann systematisch alle erreichbaren Zustände betrachtet werden, die keine Sackgassen sind (wie auch bei den anderen Bestensuchverfahren)
- Anders als gierige Bestensuche ist A* mit sicherer Heuristik auch ohne Duplikatelimination semi-vollständig, sofern alle Aktionskosten grösser als 0 sind. (Warum?)
- Dies gilt sogar für unendliche Zustandsräume (die wir hier allerdings nicht formalisiert haben), sofern jeder Zustand nur endlich viele Nachfolger hat und das Infimum aller Aktionskosten grösser als 0 ist.

7.6 Zeit- und Platzaufwand von A*

Zeitaufwand von A*

Wie hoch ist der Zeitaufwand?

- hängt stark von der Qualität der Heuristik ab
- ein Extrem: $h = 0$ für alle Zustände
  - $\rightsquigarrow$ A* identisch mit uniformer Kostensuche
- anderes Extrem: $h = h^*$ und $cost(a) > 0$ für alle Aktionen a
  - $\rightsquigarrow$ A* mit geeignetem Tie-Breaking (bevorzuge Knoten mit kleinem h) expandiert nur Knoten auf einer optimalen Lösung
  - $\rightsquigarrow$ $O(l^*)$ expandierte Knoten, $O(l^* b)$ generierte Knoten, wenn l^* Länge der gefundenen optimalen Lösung, b Verzweigungsgrad

Zeitaufwand von A*

Genauere Analysen:

- Abhängigkeit der Laufzeit vom Fehler von A*

Beispiel: Probleme mit konstantem Verzweigungsgrad und konstantem absolutem Fehler: $|h^*(s) - h(s)| \leq c$ für alle $s \in S$

- Zustandsraum ist Baum: A* skaliert linear in Lösungslänge (Pohl 1969; Gaschnig 1977)
- allgemeiner Zustandsraum: A* bleibt exponentiell in Lösungslänge (Helmert & Röger 2008)

Zeitaufwand für Reopening

Wie teuer ist Reopening?

- Für die meisten praktischen Probleme und inkonsistenten zulässigen Heuristiken werden vernachlässigbar wenige Zustände mehrfach expandiert.
- Ausnahmen existieren:
  Martelli (1977) konstruiert Zustandsräume mit n Zuständen und exponentiell vielen Reexpansionen durch A*
  (somit exponentiell schlechter als uniforme Kostensuche)

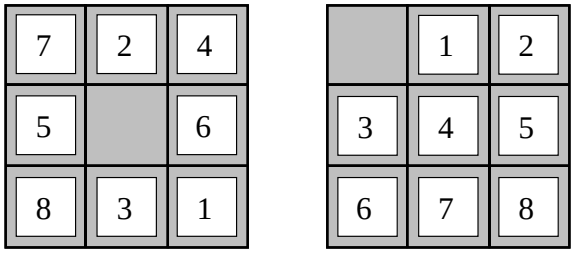
Platzaufwand von A*

Platzaufwand von A*:

- alle erzeugten Knoten werden dauerhaft gespeichert

$\rightsquigarrow$ Platzaufwand kann so gross sein wie Zeitaufwand (evtl. geringer wegen Duplikaten und Reopening)

Praktische Evaluation von A*



Start State Goal State

h_1 = Anzahl Kacheln in falscher Position (misplaced tiles)
h_2 = Summe der Abstände der Kacheln zur Zielposition (Manhattan-Distanz)

Praktische Evaluation von A*

- Jede Zeile Durchschnitt über 100 Instanzen (Anfangszustände)
- keine Duplikateliminierung

	erzeugte Knoten		
h^*	IDS	A*(h_1)	A*(h_2)
2	10	6	6
4	112	13	12
6	680	20	18
8	6384	39	25
10	47127	93	39
12	3644035	227	73
14	-	539	113
16	-	1301	211
18	-	3056	363
20	-	7276	676
22	-	18094	1219
24	-	39135	1641