

Grundlagen der Künstlichen Intelligenz

20. Klassische Suche: A*: Vollständigkeit und Komplexität

Malte Helmert

Universität Basel

30. März 2015

Klassische Suche: Überblick

Kapitelüberblick klassische Suche:

- 5.–7. Grundlagen
- 8.–12. Basisalgorithmen
- 13.–20. heuristische Algorithmen
 - 13. Heuristiken
 - 14. Analyse von Heuristiken
 - 15. Bestensuche als Graphensuche
 - 16. Gierige Bestensuche, A^* , Weighted A^*
 - 17. IDA *
 - 18. A^* : Optimalität, Teil I
 - 19. A^* : Optimalität, Teil II
 - 20. A^* : Vollständigkeit und Komplexität

Vollständigkeit

Vollständigkeit von A^*

Vollständigkeit von A^* :

- A^* ist vollständig für **sichere** Heuristiken
(wie andere Bestensuchalgorithmen als Graphensuche)
- Anders als gierige Bestensuche ist A^* mit sicheren Heuristiken immer noch **semi-vollständig**, wenn **keine** Duplikateliminierung durchgeführt wird und alle Aktionskosten grösser als 0 sind.
(Warum?)
- Das ist sogar für **unendliche Zustandsräume wahr**
(in dieser Vorlesung nicht betrachtet),
wenn jeder Zustand endlich viele Nachfolger hat
und das Infimum der Aktionskosten grösser als 0 ist.

Zeit- und Speicheraufwand von A^*

Zeitaufwand von A^* (1)

Was ist der Zeitaufwand von A^* ?

- hängt stark von der Qualität der Heuristik ab
- ein Extremfall: $h = 0$ für alle Zustände
 - ↪ A^* identisch mit uniformer Kostensuche
- ein anderer Extremfall: $h = h^*$ und $cost(a) > 0$ für alle Aktionen a
 - ↪ A^* expandiert nur Knoten entlang einer optimalen Lösung
 - ↪ $O(\ell^*)$ expandierte Knoten, $O(\ell^* b)$ erzeugte Knoten, wobei
 - ℓ^* : Länge der gefundenen optimalen Lösung
 - b : Verzweigungsgrad

Zeitaufwand von A^* (2)

Genauere Analyse:

- Abhängigkeit der Laufzeit von A^* vom **Heuristik-Fehler**

Beispiel:

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

Zeitaufwand:

- **Zustandsraum ist Baum**: Laufzeit von A^*
skaliert linear in Lösungslänge (Pohl 1969; Gaschnig 1977)
- **allgemeine Zustandsräume**: Laufzeit von A^*
skaliert exponentiell in Lösungslänge (Helmert & Röger 2008)

Zeitaufwand durch Reopening

Wie aufwändig ist Reopening?

- Für die meisten praktischen Probleme und inkonsistenten, aber zulässigen Heuristiken, ist die Zahl wiedereröffneter Knoten **vernachlässigbar**.
- **Ausnahmen** existieren:
Martelli (1977) konstruiert Zustandsräume mit n Zuständen, bei denen **exponentiell** (in n) viele Zustände von A^* wiedereröffnet werden
($\rightsquigarrow$ exponentiell schlechter als uniforme Kostensuche)

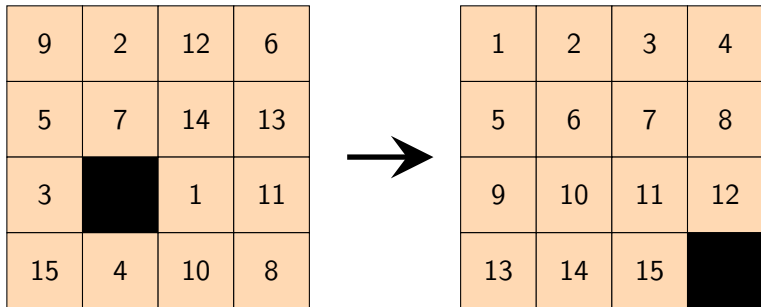
Speicheraufwand von A^*

Speicheraufwand von A^* :

- alle erzeugten Knoten werden potenziell für immer gespeichert (Ausnahme: Duplikate)
- ↪ Speicheraufwand kann so gross wie Zeitaufwand sein (potenziell niedriger wegen Duplikaten und Reopening)

Praktische Auswertung

Praktische Auswertung von A^* (1)



h_1 : Anzahl Kacheln in falscher Position (**misplaced tiles**)

h_2 : Summe der Distanzen der Kacheln zu deren Zielposition
(**Manhattan-Distanz**)

Praktische Auswertung von A* (2)

- Experimente mit zufälligen Anfangszuständen, erzeugt durch **Random Walk** vom Zielzustand
- Einträge zeigen **Median** über Anzahl der **erzeugten Knoten** für 101 Random Walks derselben Länge N

	erzeugte Knoten		
N	BFS-Graph	A* mit h_1	A* mit h_2
10	63	15	15
20	1'052	28	27
30	7'546	77	42
40	72'768	227	64
50	359'298	422	83
60	> 1'000'000	7'100	307
70	> 1'000'000	12'769	377
80	> 1'000'000	62'583	849
90	> 1'000'000	162'035	1'522
100	> 1'000'000	690'497	4'964

Zusammenfassung

Zusammenfassung (1)

Weitere Eigenschaften von A^* :

- **Vollständigkeit:**
 - **vollständig** für sichere Heuristiken
 - **semi-vollständig**, wenn als Baumsuche statt Graphensuche und Aktionskosten positiv
 - **semi-vollständig** für unendliche Zustandsräume mit endlicher Verzweigung, wenn Infimum der Aktionskosten positiv

Zusammenfassung (2)

Weitere Eigenschaften von A^* :

- **Zeitkomplexität:**
 - Null-Heuristik: wie uniforme Kostensuche
 - perfekte Heuristik, positive Aktionskosten:
nur optimale Lösung wird expandiert
 - allgemein: abhängig von Qualität der Heuristik
und Graphstruktur des Zustandsraums
 - Reopening normalerweise vernachlässigbar;
in degenerierten Fällen extrem teuer
- **Speicherkomplexität:** ähnlich Zeitkomplexität