Jun.-Prof. Dr. Robert Jäschke, Gerhard Gossen
Fachgebiet Wissensbasierte Systeme

12.05.2015

4. Übung „Künstliche Intelligenz“

Sommersemester 2015

1 Suche

1. Berechnen Sie mit dem A*-Algorithmus und mit dem Greedy-Algorithmus einen optimalen Weg von Arad nach Bucharest. Protokollieren Sie die einzelnen Stufen wie gewohnt tabellarisch. Verwenden Sie dazu wie bisher die in Abb. 1 gezeigte Strassenkarte sowie die in Abb. 2 gezeigte Heuristik.

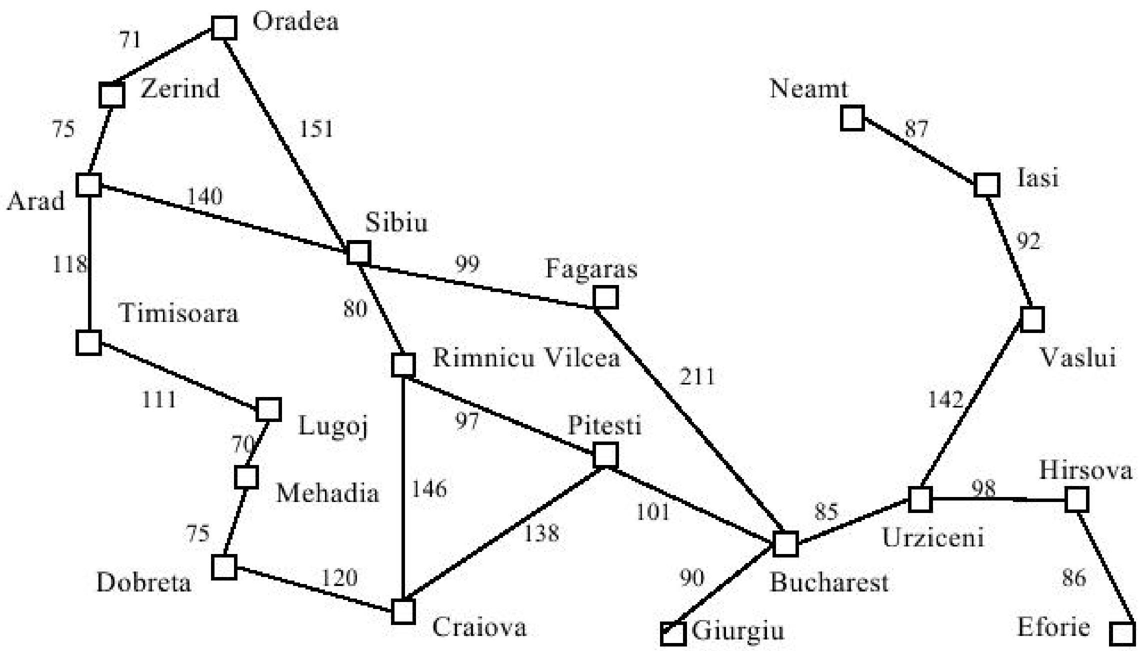

Abbildung 1: Eine vereinfachte Straßenkarte eines Teils von Rumänien

Arad	366	Mehdia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Dobreta	242	Pitesti	100
Eforie	161	Riminicu Vilcea	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zering	374

Abbildung 2: Werte der Heuristik-Luftliniendistanz zu Bucharest

Musterlösung:

A*-Algorithmus

Schritt	OPEN	CLOSED
1	Arad	
2	Sibiu(140+253=393), Timisoara(118+329=447), Zerind(75+374=449)	Arad
3	Rimnicu Vilcea(140+80+193=413), Fagaras(140+99+176=415), Timisoara(447), Zerind(449), Oradea(291+380=671)	Arad, Sibiu
4	Fagaras(415), Pitesti(220+97+100=417), Timisoara(447), Zerind(449), Craiova(220+146+160=526), Oradea(671)	Arad, Sibiu, Rimnicu Vilcea
5	Pitesti(417), Timisoara(447), Zerind(449), Bucharest(239+211+0=450), Craiova(526), Oradea(671)	Arad, Sibiu, Rimnicu Vilcea, Fagaras
6	Bucharest, Timisoara, Zerind, Craiova, Oradea	Arad, Sibiu, Rimnicu Vilcea, Fagaras, Pitesti

Greedy

Schritt	OPEN	CLOSED
1	Arad	
2	Sibiu(253), Timisoara(329), Zerind(374)	Arad
3	Fagaras(176), Rimnicu Vilcea(193), Timisoara(329), Zerind (374)	Arad, Sibiu
4	Bucharest(0), Rimnicu Vilcea, Timisoara, Zerind	Arad, Sibiu, Fagaras

2. Vergleichen Sie die Ergebnisse, welche Breiten- und Tiefensuche, Greedy, A* und der Bergsteigeralgorithmus geliefert haben. Vergleichen Sie insbesondere Greedy und den Bergsteigeralgorithmus.

Musterlösung:

Der A*-Algorithmus und Greedy berechnen für das gegebene Problem dieselben Mengen CLOSED, OPEN und SUCC, da für alle Zustände (Orte) gilt, daß sie umso früher in OPEN landen, je weiter sie von Bucharest laut der Heuristik entfernt sind. Daher gilt für alle Elemente in SUCC(z), daß ihre Bewertung durch die Heuristik kleiner ist, als die der Elemente in OPEN, die in der Reihenfolge nach z vor seiner Expansion stehen. Durch das Sortieren über OPEN und nicht nur über SUCC(z), ändert sich also die Reihenfolge nicht. Für die Schritte zählen wir lediglich die Anzahl an Expansionen. Die Komplexität zählt, wieviele Datenobjekte angefaßt wurden: Eine Expansion kostet $|SUCC(z)|$, das Sortieren einer Menge M (SUCC(z) im Fall des Bergsteiger- und OPEN für den Greedy-Algorithmus) kostet $\lceil |M| \log_2(|M|) \rceil$.

Strategie	Schritte	Komplexität	Optimal
Tiefensuche	4	7	i.d.R. nein
Breitensuche	7	8	nein
Greedy	4	7+11	nein
Bergsteiger	4	8+6	nein
A*	6	9+28	ja

Die Optimalität der Tiefensuche hängt davon ab, ob man per Zufall den günstigsten Weg gewählt hat, wovon man im Allgemeinen aber nicht ausgehen kann.

3. Das Problem des *Schachspielens* kann als Suchproblem interpretiert werden.

   Listen Sie umgangssprachlich die Zustände, Operatoren, Startzustände und mögliche Endzustände auf.

Musterlösung:

Startzustand: Initialbelegung des Schachbretts als ein Startzustand.
mögliche Endzustände: alle Figurenkombinationen in denen ein König schachmatt ist.
Operatoren: alle möglichen Schachzüge die ein Schachspieler anwenden kann.
Zustände: alle Figurenkombinationen.

4. Wie unterscheiden sich die Lösungsstrategien bei den 2 vorhergehenden Aufgabenstellungen (optimale Wege, Schachspiel) voneinander?

Musterlösung:

Beim Schachspiel ist ein (menschlicher) Gegner involviert, dessen Reaktionen (Züge) beachtet werden müssen. Möchte man eine Strategie zum Gewinnen eines Spiels berechnen, so muss man alle möglichen Züge des Gegners in Betracht ziehen. Die Lösung besteht daher aus einem vielfach verzweigten Baum – wobei die Knoten, die die Entscheidungen der Maschine darstellen, keine Verzweigungen haben (die Maschine wählt den optimalen Zug).

5. Macht es für Schachspiel die bidirektionale Suche zu verwenden? Begründen sie ihre Antwort.

Musterlösung:

Nein, denn beim Schachspiel gibt es mehrere Gewinnsituationen – die Anzahl der möglichen Gewinnstellungen ist sehr groß.

6. Formulieren Sie das *8-Damen-Problem*. Beschreiben Sie umgangssprachlich mögliche Zustände, den Ausgangszustand, eine Nachfolgerfunktion und den Zielzustand. Wie lässt sich der Suchraum passend einschränken? Beschreiben Sie ebenfalls Zustände und Nachfolgerfunktion.

Musterlösung:

Inkrementelle Formulierung (beginnend mit einem leeren Schachbrett):

Zustände: Eine beliebige Anordnung von 0 bis 8 Damen auf dem Brett.

Ausgangszustand: Ein leeres Schachbrett.

Nachfolgerfunktion: Einem beliebigen Quadrat eine Dame hinzufügen.

Zielzustand: 8 Damen befinden sich auf dem Schachbrett und keine wird angegriffen

Bei dieser Formulierung müssen $64x63x...57$ Abfolgen ausgewertet werden. Eine bessere Formulierung ist eine vollständige Zustandsformulierung:

Zustände: Anordnung von n Damen $0 \leq n \leq 8$, eine pro Spalte in den am weitesten links liegenden n Spalten, wobei keine Dame eine andere angreift, sind Zustände.

Nachfolgerfunktion: Einem beliebigen Quadrat in der am weitesten links liegenden leeren Spalte wird eine Dame hinzugefügt, so dass diese nicht durch eine andere Dame angegriffen wird.

Diese Lösung reduziert den Zustandsraum auf nur 2057 Zustände. Eine zulässige Lösung ist im Folgenden zu finden.