

3. Übung „Künstliche Intelligenz“

Sommersemester 2015

Suche

1. Das abgerollte Streckennetz, das Sie in der letzten Übung erstellt haben, kann man auch als Zustands- oder Entscheidungsgraphen eines Suchproblems interpretieren. Diskutieren Sie die bei einer Tiefensuche auftretenden Probleme. Wie muß das Schema (Kapitel 2, Folie 11) angepaßt werden, damit die Probleme umgangen werden können?

Musterlösung:

Im letzten Schritt S4 muß NEW(z) entstehen aus SUCC(z) durch Löschen aller Zustände in CLOSED.

2. In der letzten Vorlesung wurde das Thema *Suche* besprochen. Welche möglichen Anforderungen erfüllen die Tiefensuche und die Breitensuche? Welche Lösungen liefern die beiden Verfahren? Beschreiben Sie die Vorgehensweise beider Verfahren sowie jeweilige Vor- und Nachteile.

Musterlösung:

Tiefensuche: Die Suchschlange ist nach dem last-in/first-out-Prinzip („LIFO“ bzw. „Stack“) organisiert. Es wird diejenige Lösung gefunden, die sich in der Tiefe des Suchbaumes befindet. Bei unendlichen Pfaden werden nicht alle Lösungen geliefert. Probleme bereitet die Tiefensuche, wenn eine unendliche Anzahl von erreichbaren Zuständen vorliegt. Das Traversieren des Suchbaumes ist dann unmöglich. Allerdings lässt sich die Tiefensuche speichereffizient rekursiv implementieren, da man anstatt einer Suchschlange einen Stack verwenden kann.

Breitensuche: Die Suchschlange der Breitensuche ist nach dem first-in/first-out Prinzip (FIFO) organisiert. Innerhalb des Suchbaums wird der kürzeste Weg gefunden. Jedoch benötigt die Breitensuche im Vergleich zur Tiefensuche viel Speicherplatz $\theta(b^d)$, da die Zahl der Knoten exponentiell mit der Baumtiefe steigt.

Allgemein: Die Breitensuche ist vollständig, da jede Schicht des Suchbaumes systematisch durchsucht wird und jeder Knoten expandiert wird. Die Tiefensuche kann hingegen unvollständig sein, wenn der Suchbaum einen unendlichen Ast besitzt. Dadurch terminiert die Tiefensuche nicht.

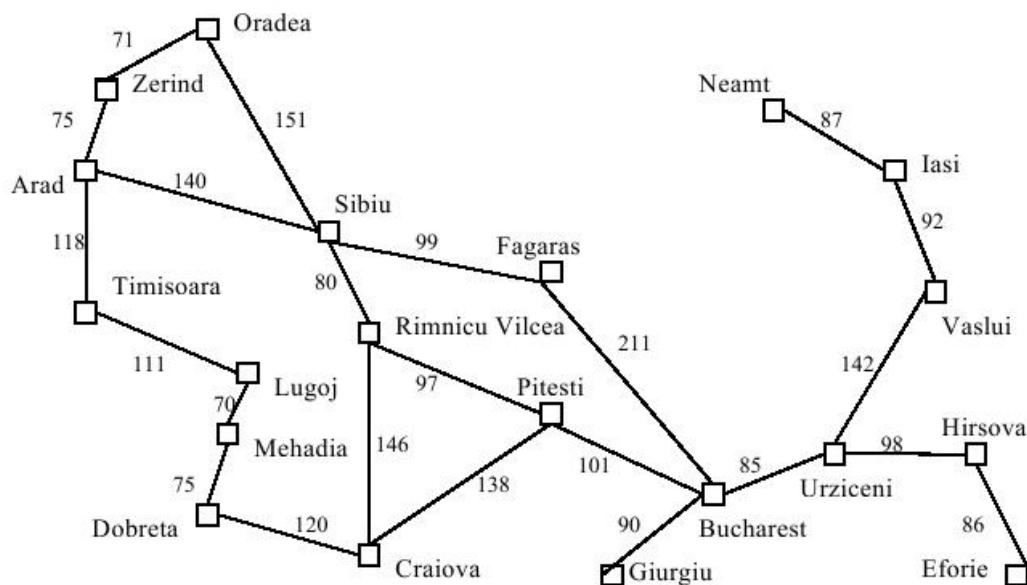


Abbildung 1: Eine vereinfachte Straßenkarte eines Teils von Rumänien

3. Wie muß der Schritt S4 im Schema zur Suche (Kapitel 2, Folie 11) angepaßt werden, um Tiefen- oder Breitensuche zu realisieren?

Musterlösung:

Für die Tiefensuche muß $NEW(z)$ am Anfang von OPEN eingefügt werden, für die Breitensuche muß die Menge ans Ende gestellt werden.

4. Suchen Sie jeweils mit dem Tiefen-, Breitensuche- und optimistischen Bergsteiger-Algorithmus einen Weg von Arad nach Bucharest. Verwenden Sie das Schema, welches in den Vorlesungsunterlagen (Kapitel 2, Folie 11/18) zu finden ist. Protokollieren Sie die einzelnen Stufen tabellarisch. Benutzen Sie folgendes Tabellenschema zum Protokollieren:

Schritt	OPEN	CLOSED	SUCC

Dabei steht in den Spalten OPEN und CLOSED jeweils die entsprechende Menge zu Beginn des Schrittes, in der Spalte SUCC die während des Schritts gebildete Menge.

Verwenden Sie die in Abb. 1 gezeigte Strassenkarte sowie die in Tab. 2 gezeigte Heuristik.

Musterlösung:

a) Tiefensuche

Schritt	OPEN	CLOSED	SUCC
1	Arad		Sibiu, Timisoara, Zerind
2	Sibiu, Timisoara, Zerind	Arad	Fagaras, Oradea, Rimnicu V.
3	Fagaras, Oradea, Rimnicu Vilcea, Timisoara, Zerind	Arad, Sibiu	Bucharest

b) Breitensuche

Schritt	OPEN	CLOSED	SUCC
1	Arad		Timisoara, Sibiu, Zerind
2	Sibiu, Timisoara, Zerind	Arad	Fagaras, Oradea, Rimnicu V.
3	Timisoara, Zerind, Fagaras, Oradea, Rimnicu Vilcea	Arad, Sibiu	Lugoj
4	Zerind, Fagaras, Oradea, Rimnicu Vilcea, Lugoj	Arad, Sibiu, Timisoara	
5	Fagaras, Oradea, Rimnicu Vilcea, Lugoj	Arad, Sibiu, Timisoara, Zerind	Bucharest

c) Optimistischer Bergsteiger

Schritt	OPEN	CLOSED	SUCC
1	Arad		Zerind(374), Sibiu(253), Timisoara(329)
2	Sibiu(253)	Arad	Oradea(380), Fagaras(176), Rimnicu Vilcea(193)
3	Fagaras(176)	Arad, Sibiu	Bucharest(0)

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

Tabelle 2: Werte der Heuristik-Luftliniendistanz zu Bucharest