

Fachhochschule Köln  
Cologne University of Applied Sciences

Campus Gummersbach

# Algorithmik Praktikum

Informatik

Wintersemester 2014/2015

## Der Bellman-Ford-Algorithmus

Niklas Gruhn 11097792 (niklas.gruhn@smail.fh-koeln.de)

Daniel Hammers 11092958 (daniel.hammers@smail.fh-koeln.de)

Tom Peetz 11098016 (tom.peetz@smail.fh-koeln.de)

Jan Philipp Sominka 11082254 (jan.sominka@smail.fh-koeln.de)

### **Projektleiter:**

Prof. Dr. Heiner Klocke  
Fakultät für Informatik  
und Ingenieurwissenschaften

### **Zusammenfassung**

In vielen Anwendungsbereichen kann es erforderlich sein, den günstigsten Weg zwischen zwei Punkten zu finden. Hierbei sollen aber nicht nur Entfernung von Start und Ziel, sondern vielmehr die individuellen Reisekosten inklusive negativer Reisekosten als Thematik behandelt werden. Diese Dokumentation beschäftigt sich im Allgemeinen mit dem Bellman-Ford-Algorithmus, der nach seinen Erfindern Richard Bellman und Lester Ford benannt wurde. Da auch Edward Moore zu seiner Entwicklung beigetragen hat, wird gelegentlich auch vom Moore-Bellman-Ford-Algorithmus gesprochen. Zudem stellt sie ein Programm zur Lösung vor.

## Inhaltsverzeichnis

<b>1</b>	<b>Allgemeines</b>	<b>3</b>
1.1	Der Bellman-Ford-Algorithmus . . . . .	3
1.2	Einordnung des Algorithmus . . . . .	3
1.3	Praktische Anwendung . . . . .	3
1.4	Vergleich zu Dijkstra . . . . .	3
<b>2</b>	<b>Funktionsweise und Anwendungsbeispiel</b>	<b>4</b>
2.1	Funktionsweise . . . . .	4
2.2	Laufzeiten . . . . .	5
2.3	Pseudocode . . . . .	6
<b>3</b>	<b>Multiple Choice Fragen(Richtige Antworten unterstrichen)</b>	<b>7</b>
<b>4</b>	<b>Quellen- und Literaturverzeichnis</b>	<b>8</b>

# 1 Allgemeines

Im Folgenden Abschnitt beschäftigen wir uns zunächst mit der Idee des Algorithmus, und gehen später auf praktische Anwendungsbeispiele ein.

## 1.1 Der Bellman-Ford-Algorithmus

Der Bellman-Ford-Algorithmus dient der Suche nach dem Kürzesten Weg in einem kantengewichteten Graphen zwischen einem Start- und einem Endknoten. Ist die niedrigste mögliche Summe der zugehörigen Kanten vom Start- bis zum Zielpunkt gefunden, so wurde der kürzeste Weg bestimmt. Der Graph muss hierbei gerichtet sein und darf sowohl positive, als auch negative Gewichtung besitzen. Als Rückgabe wird ein Boolescher Wert zurückgegeben, wenn der Algorithmus einen negativen Zyklus in dem Graphen entdeckt. Wenn dies nicht der Fall ist, bestimmt er die kürzesten Pfade. Der Bellman-Ford Algorithmus arbeitet dynamisch nach dem Optimalitätsprinzip von Bellman.

## 1.2 Einordnung des Algorithmus

Der Bellman-Ford Algorithmus arbeitet dynamisch nach dem Optimalitätsprinzip von Bellman, welches besagt, dass sich eine Optimallösung aus optimalen Teillösungen zusammensetzt. Nach diesem Prinzip muss ein kürzester Weg S zwischen dem Startknoten A und dem Zielknoten B, der durch die Knoten X und Y führt auch zwischen X und Y den kürzesten Weg verwenden.

## 1.3 Praktische Anwendung

Praktisch angewendet wird der Algorithmus im Distanzvektoralgorithmus um Routingtabellen anzupassen. Ändert sich beispielsweise die Netzwerktopologie, so wird dies erkannt und es werden mittels des Bellman-Ford-Algorithmus neue Routen zu den betroffenen Knoten berechnet. Diese Änderung in der Routingtabelle wird an alle erreichbaren Knoten gesendet, die Ihr Routing hierdurch ebenfalls anpassen und die Änderung ihrerseits weiter propagieren.

## 1.4 Vergleich zu Dijkstra

Während bei Bellman-Ford ein gerichteter Graph als Voraussetzung gilt, ist es bei Dijkstra sowohl mit ungerichteten als auch gerichteten Graphen möglich. Dijkstra kann nur mit positiven Gewichtungen umgehen, wo hingegen Bellman-Ford mit beiden umgehen kann. Der Bellman-Ford Algorithmus benutzt keine Priority-queues, da ein günstigerer weg durch einen teureren Knoten führen könnte.

## 2 Funktionsweise und Anwendungsbeispiel

Im Folgenden werden wir die genaue Funktionsweise erläutern, sowie einen Konkreten Fall anhand eines Beispiels lösen.

### 2.1 Funktionsweise

Abbildung 1: Der Algorithmus geht am Anfang von einer sehr schlechten Schätzung jedes Knoten aus. Dies bedeutet, dass jeder Knoten, außer dem Startknoten, unendliche Kosten hat.

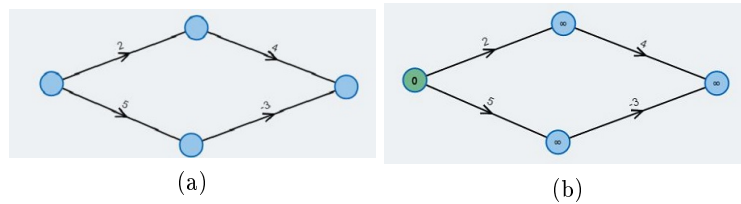


Abbildung 2: Ein Durchlauf wird Phase genannt, wir beginnen mit Phase 1 wo nur Wege mit einer Kante betrachtet werden. Bei jedem weiteren Durchlauf kommt eine Kante hinzu, bis alle Wege betrachtet wurden.

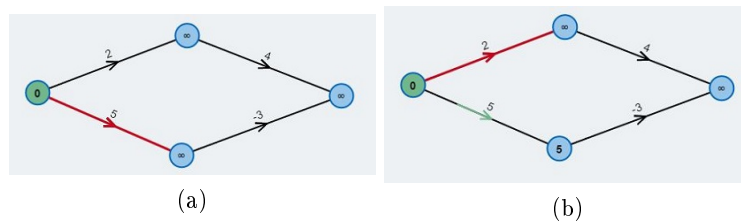


Abbildung 3: In jeder Phase wird nun berechnet, ob die Kosten des Startknotens plus Kante niedriger sind, als die kosten des Zielknotens.

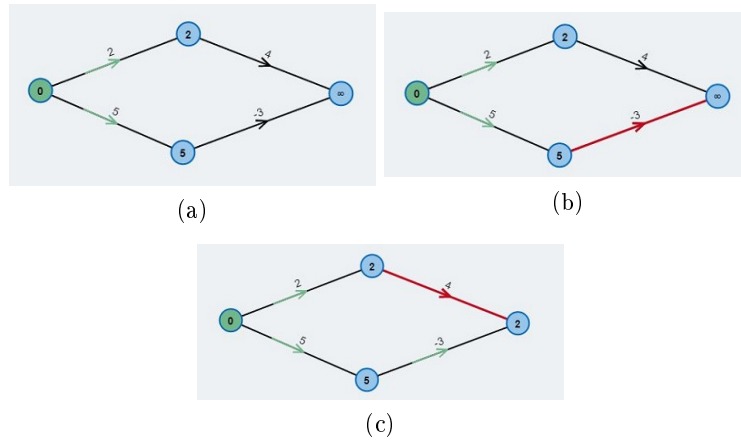
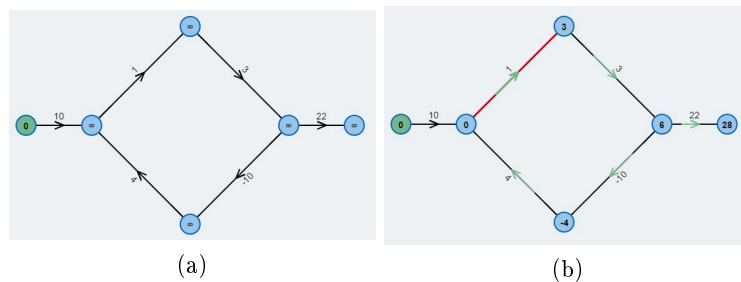


Abbildung 4: Falls dies der Fall ist, wurde ein besserer Weg als der bisher bekannte gefunden. Wenn alle Wege betrachtet wurden, wird überprüft ob es einen negativen Zyklus gibt. Dafür wird überprüft, ob die bisherigen Kosten des Zielknotens höher sind, als die Kosten des Anfangsknotens addiert mit der Kante. Wenn dies der Fall ist, so existiert ein negativer Zyklus in dem Graphen.



(c) Quelle der Abbildungen: <http://www-m9.ma.tum.de/material/spp/Bellman-Ford-Algorithmus.html>

## 2.2 Laufzeiten

Da die Laufzeit direkt abhängig von der Anzahl der Knoten und Kanten ist, sieht die Berechenbarkeit wie folgt aus:

Laufzeit Gesamt:  $O(V E)$   
 Initialisierung:  $\Theta(V)$   
 Durchläufe über die Kanten:  $\Theta(E)$   
 Die for-schleife:  $O(E)$

## 2.3 Pseudocode

Im Pseudocode sieht dieses Verfahren wie folgt aus:

```
01  für jedes v aus V    \\
02      Distanz(v) := unendlich, Vorgänger(v) := keiner\\
03  Distanz(s) := 0\\
04  wiederhole n - 1 mal    \\
05      für jedes (u,v) aus E\\
06          wenn Distanz(u) + Gewicht(u,v) < Distanz(v)\\
07              dann\\
08                  Distanz(v) := Distanz(u) + Gewicht(u,v)\\
09                  Vorgänger(v) := u\\
10  für jedes (u,v) aus E    \\
11      wenn Distanz(u) + Gewicht(u,v) < Distanz(v) dann\\
12          STOPP mit Ausgabe "Es gibt einen Zyklus negativen Gewichtes."\\
13  Ausgabe Distanz\\
```

Quelle: <http://de.wikipedia.org/wiki/Bellman-Ford-Algorithmus>

### 3 Multiple Choice Fragen(Richtige Antworten unterstrichen)

01. Zu welcher Klasse gehört der Bellman-Ford-Algorithmus?

- a) Greedy
- b) Approximation
- c) Dynamisch
- d) Evolutionär

02. Wie bewertet der Algorithmus bei der Initialisierung die einzelnen Knoten außer dem Startknoten?

- a) n
- b) 0
- c)  $\infty$
- d)  $n \cdot m - 1$

03. Welche der folgenden Angaben wird zur Ausführung nicht benötigt?

- a) Kosten der Knoten
- b) Endknoten
- c) Startknoten
- d) Gewicht der Kanten

04. Welche Laufzeit hat der Bellman-Ford-Algorithmus?

- a)  $O(V \cdot E)$
- b)  $O(V)$
- c)  $O(E)$
- d)  $O(V^2)$

05. Wie geht der Algorithmus mit Negativzyklen um?

- a) Er Gibt einen Booleschen wert zurück
- b) Er stürzt ab
- c) Er gibt die Kürzesten Wege zu allen Knoten zurück
- d) Es können keine Negativzyklen vorkommen.

06. Welche Gewichtung dürfen die Kanten besitzen?

- a) Nur positive Gewichtung
- b) Nur negative Gewichtung
- c) Positive und negative Gewichtung
- d) Die Kanten haben keine Gewichtung

07. Mit welcher Art von Graphen kann der Algorithmus umgehen?

- a) Ungerichteter Graph
- b) Gerichteter Graph
- c) Gerichteter und ungerichteter Graph
- d) Die Art des Graphen ist uninteressant für den Algorithmus



## 4 Quellen- und Literaturverzeichnis

-[http : //bibiserv.techfak.uni – bielefeld.de/adp/ps/adp\\_discipline.pdf](http://bibiserv.techfak.uni-bielefeld.de/adp/ps/adp_discipline.pdf)(zuletzt besucht: 28.10.2014)

-Sven Oliver Krumke und Hartmut Noltemeier: Graphentheoretische Konzepte und Algorithmen. 2. Auflage Vieweg-Teubner 2009. ISBN 978-3-8348-0629-1

-[http : //www-m9.ma.tum.de/material/spp/Bellman-Ford-Algorithmus.html](http://www-m9.ma.tum.de/material/spp/Bellman-Ford-Algorithmus.html)(zuletzt besucht: 28.10.2014)

- Thomas H. Cormen, Charles Leiserson, Ronald L. Rivest, Clifford Stein: Algorithmen – Eine Einführung. 2. Auflage. Oldenbourg Wissenschaftsverlag, München 2007, ISBN 978-3-486-58262-8

-[http : //th0ms0n.no-ip.org/blog/bellmanford.html](http://th0ms0n.no-ip.org/blog/bellmanford.html)(zuletzt besucht: 28.10.2014)

- Andrew S. Tanenbaum: Computernetzwerke, Pearson Studium, 4. überarbeitete Auflage (15. Juli 2003), ISBN 978-3827370464