



Technische
Universität
Braunschweig

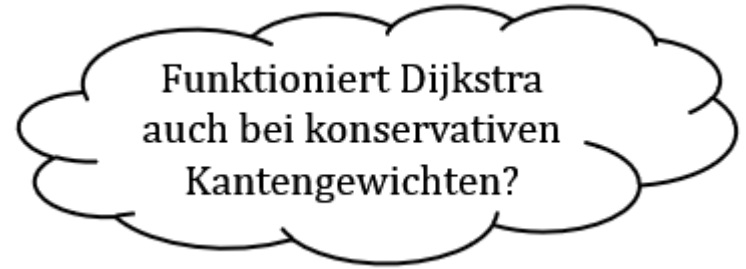
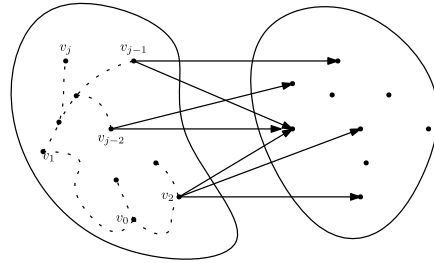
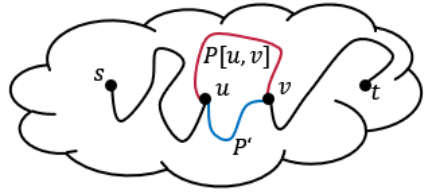


Netzwerkalgorithmen – Vorlesung #5

Arne Schmidt

Wiederholung

Wiederholung



Algorithmus 3.6

Eingabe:

Digraph $D = (V, A)$, Kostenfunktion $c: A \rightarrow \mathbb{R}_{\geq 0}$, Startknoten $s \in V$

Ausgabe:

Länge $\ell(v)$ eines kürzesten sv -Pfades, Vorgänger $p(v)$ von v auf diesem Pfad

1. Function $\text{DIJKSTRA}(D, c, s)$

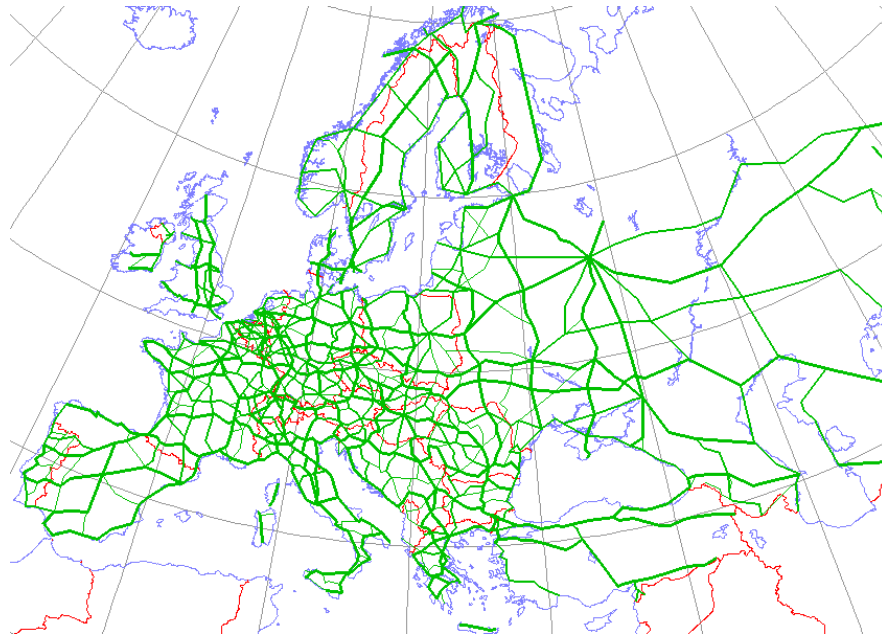


Satz 3.8

Dijkstra's Algorithmus löst Problem 3.4 (Kürzeste Wege) korrekt für nicht-negative Kantenkosten in $O(n \log n + m)$ Zeit.

Exkurs – Fast and Furious

Straßennetze



Internationales Europastraßennetz

- Straßennetze in der Regel dünn, d.h. $O(n)$ Kanten
- Knoten besitzen Koordinaten
- Kantenkosten > 0

Problem 3.1: Kürzester Weg Problem (Shortest Path, SP)

Gegeben:

Digraph $D = (V, A)$

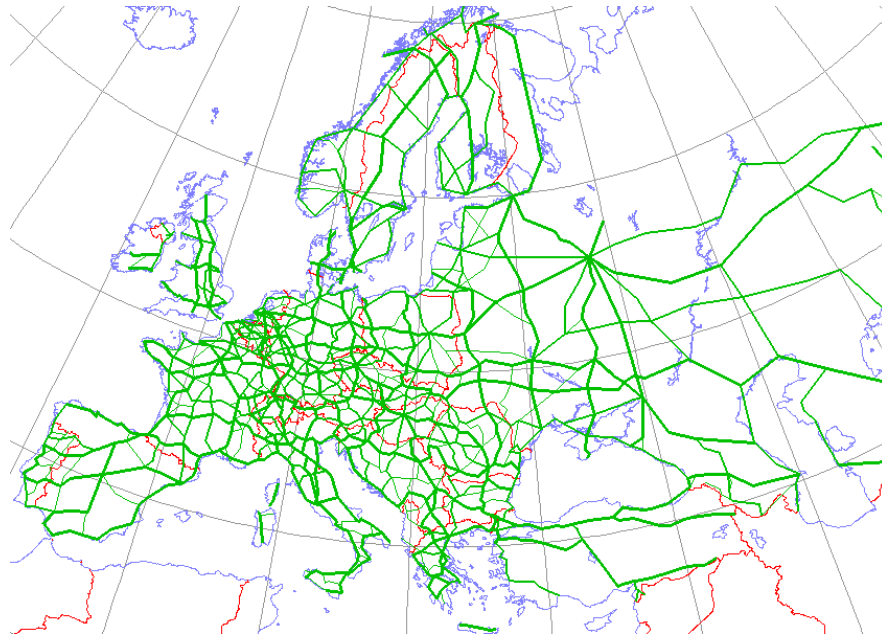
Kostenfunktion $c: A \rightarrow \mathbb{R}$

Knoten $s, t \in V$

Gesucht:

Ein kostenminimaler st -Weg

Straßennetze



Internationales Europastraßennetz

- Straßennetze ändern sich nicht oft.
- U.U. viele Anfragen („Queries“) zu bearbeiten.
- Nutze Preprocessing, um schnell zu sein.



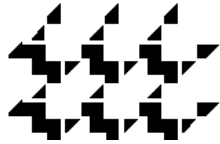
$O(n^2 \log n)$ Laufzeit
 $O(n^2)$ Speicherbedarf

Algorithm Engineering

2005/2006

DIMACS

Center for Discrete Mathematics & Theoretical Computer Science
Founded as a National Science Foundation Science and
Technology Center

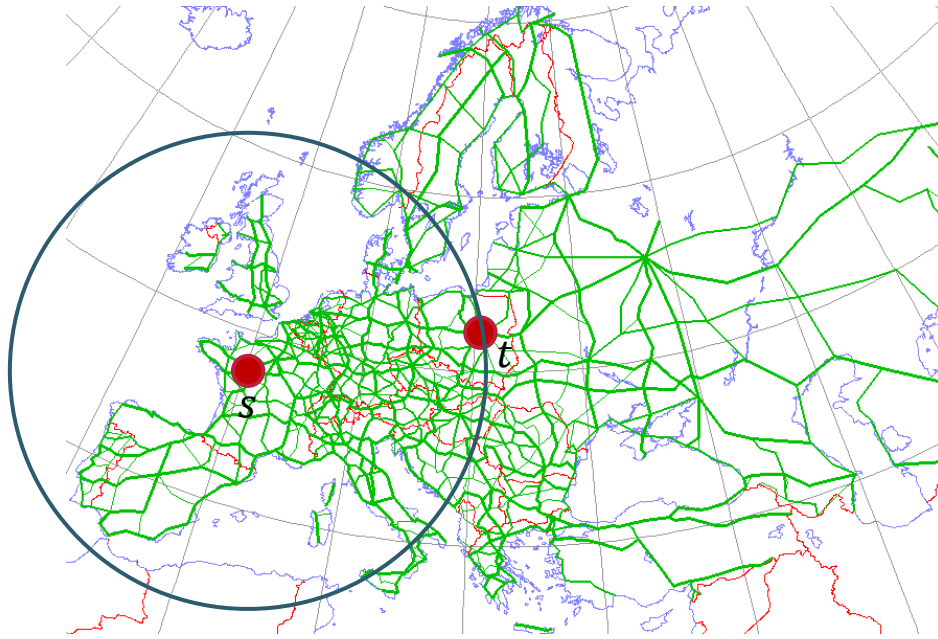


n Mal naiv Dijkstra dauert Tage!

9th DIMACS Implementation Challenge

Name	Description	# nodes	# arcs
USA	Full USA	23,947,347	58,333,344
CTR	Central USA	14,081,816	34,292,496
W	Western USA	6,262,104	15,248,146
E	Eastern USA	3,598,623	8,778,114
LKS	Great Lakes	2,758,119	6,885,658
CAL	California and Nevada	1,890,815	4,657,742
NE	Northeast USA	1,524,453	3,897,636
NW	Northwest USA	1,207,945	2,840,208
FLA	Florida	1,070,376	2,712,798
COL	Colorado	435,666	1,057,066
BAY	San Francisco Bay Area	321,270	800,172
NY	New York City	264,346	733,846

Dijkstra

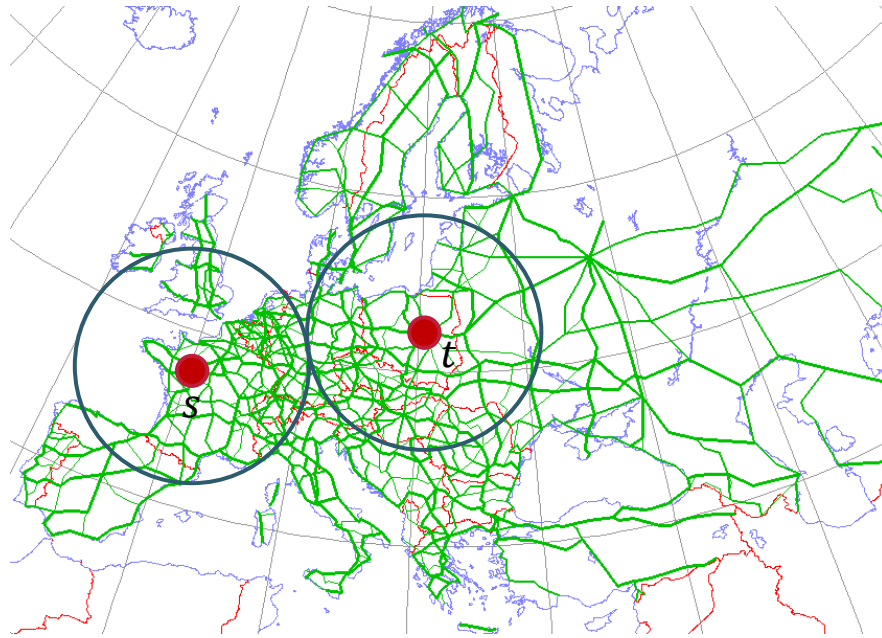


Internationales Europastraßennetz

- R breitet sich kreisförmig aus
- $|R|$ wächst quadratisch im Radius



Bidirektionaler Dijkstra

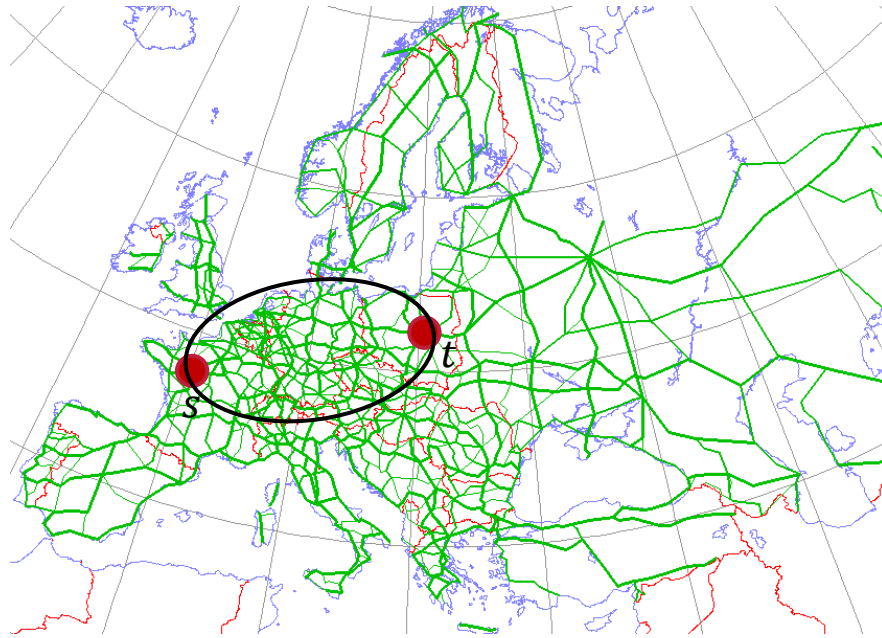


Internationales Europastraßennetz

- R breitet sich von s und t kreisförmig aus
- $|R|$ wächst weiterhin quadratisch im Radius



Heuristische Ansatz

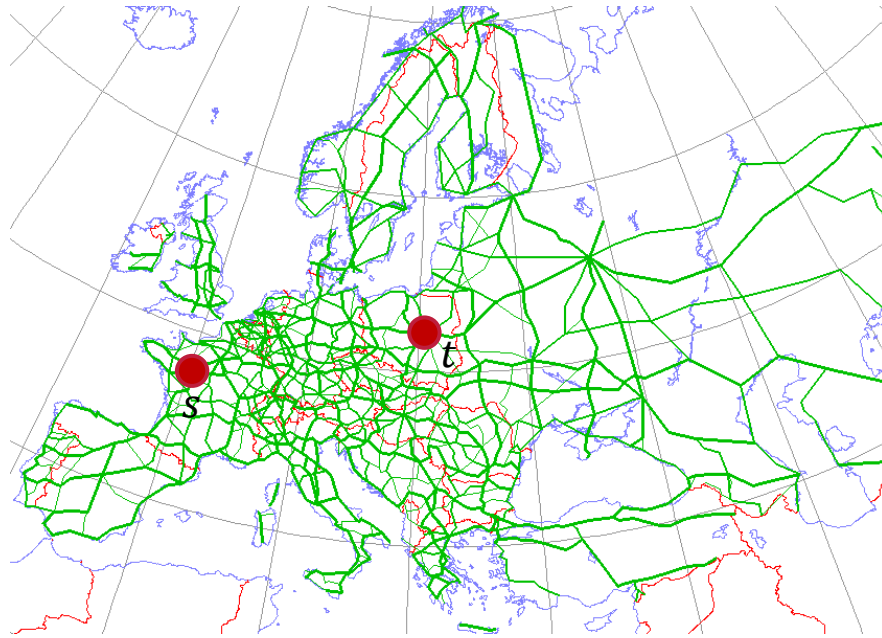


Internationales Europastraßennetz

- Gib grobe Richtung an
 - Landmark-basierend (ALT)
 - \Rightarrow A* Algorithmus (siehe gr. Übung)



Highway Hierarchies <http://algo2.iti.kit.edu/schultes/hwy/hwyHierarchiesIncomplete.pdf>



Internationales Europastraßennetz

- Unterteile Straßennetz in Straßenklassen
 - Stadtstraße
 - Kreisstraße
 - Landstraße
 - Bundesstraße
 - Autobahn
 - Europastraße
 - ...

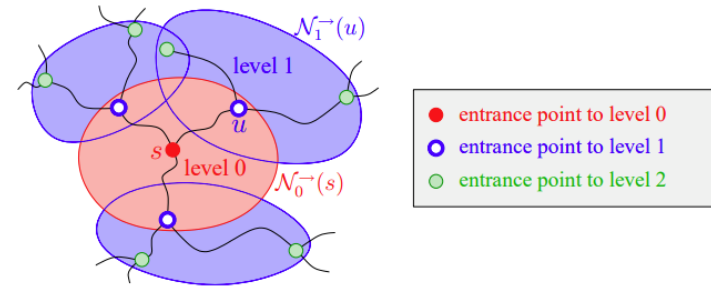


Figure 3.8: A schematic diagram of a highway query. Only the forward search is depicted.

Highway Hierarchies <http://algo2.iti.kit.edu/schultes/hwy/hwyHierarchiesIncomplete.pdf>

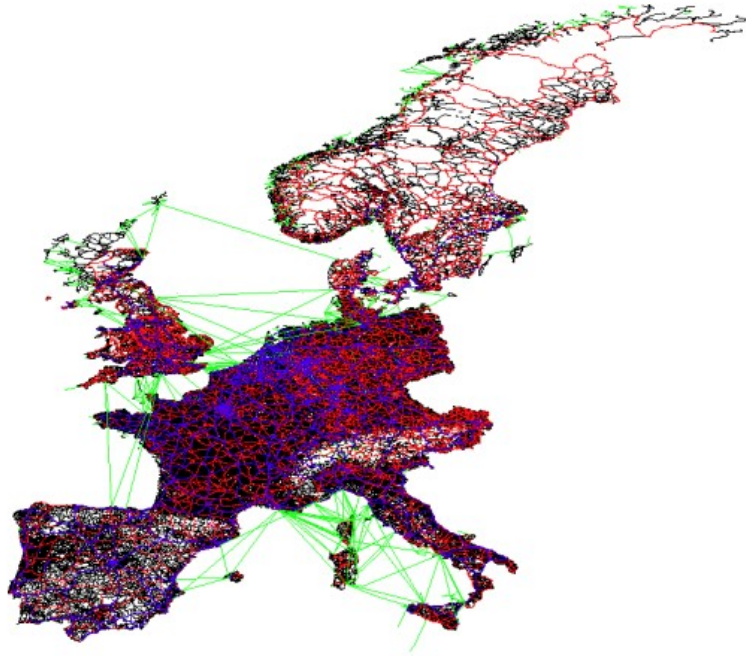
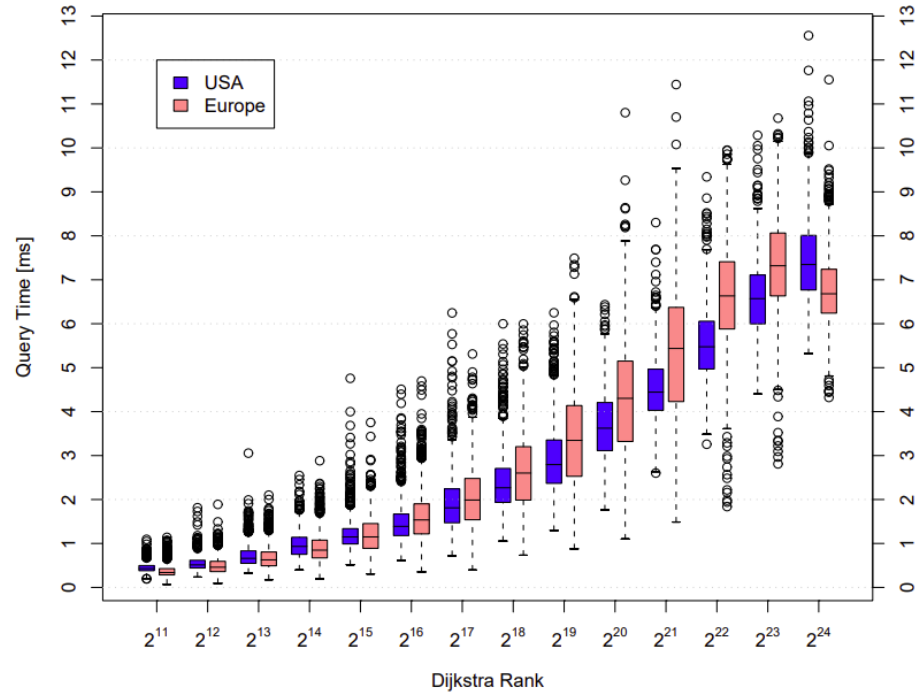


Figure 6.2: Road network of Western Europe. The colours indicate the road category: ferry, motorway, national road, regional road. The slowest category (urban street) has been omitted in this figure.



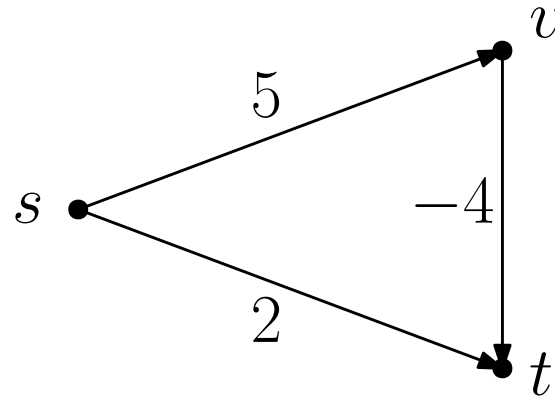
Gotta go fast! <https://i11www.iti.kit.edu/extra/publications/dssw-erpa-09.pdf>

method	space [B/n]	preproc. [min]	speedup
DIJKSTRA	21	0	1
separator multi-level	?	> 5 400	52
edge flags (basic)	35	299	523
landmark A^*	89	13	28
edge flags	30	1 028	3 951
HHs (basic)	49	161	2 645
reach + shortc. + A^*	100	1 625	1 559
HHs + dist. tab.	56	141	3 932
HHs + dist. tab. + A^*	68	13	10 364
HHs + dist. tab. + A^*	92	14	12 902
high-perf. multi-level	181	1 440	401 109
transit nodes (eco)	140	25	574 727
transit nodes (gen)	267	75	1 470 231
highway nodes	28	15	7 437
approx. planar $\epsilon = 0.01$	2 000	150	18 057
SHARC	34	107	21 800
bidirectional SHARC	41	212	97 261
contr. hier. (aggr)	17	32	41 051
contr. hier. (eco)	21	10	28 350
CH + edge flags (aggr)	32	99	371 882
CH + edge flags (eco)	20	32	143 682
transit nodes + edge flags	341	229	3 327 372
contr. hier. (mobile)	8	31	9 878

3.2.2 Konservative Kostenfunktion

Konservative Kantengewichte

Funktioniert Dijkstra
auch bei konservativen
Kantengewichten?



Algorithmus Idee

Für alle $v_k \in V \setminus R$ ist:

$$\ell_k \leq \min_{v_i \in R} (\ell_i + c((i, k)))$$

Für alle $v_k \in V$ ist:

$$\ell_k = \min_{(v_i, v_k) \in A} (\ell_i + c((i, k)))$$

Problem:

Label ℓ_i sind nicht bekannt.
⇒ Label schrumpfen lassen!

Update-Regel:

Aktualisiere Label ℓ_k über Kante $(v_i, v_k) \in A$, falls $\ell_k > \ell_i + c((i, k))$.

Frage:

Wann ist ℓ_k kleinstmöglich?

Algorithmus von Moore, Bellman und Ford

Algorithmus 3.9

Eingabe:

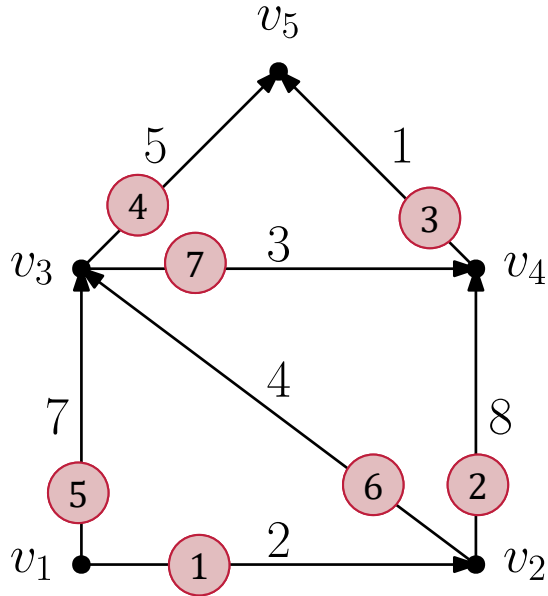
Digraph $D = (V, A)$, konservative Kostenfunktion $c: A \rightarrow \mathbb{R}$, Startknoten $s \in V$

Ausgabe:

Länge $\ell(v)$ eines kürzesten sv -Pfades, Vorgänger $p(v)$ von v auf diesem Pfad

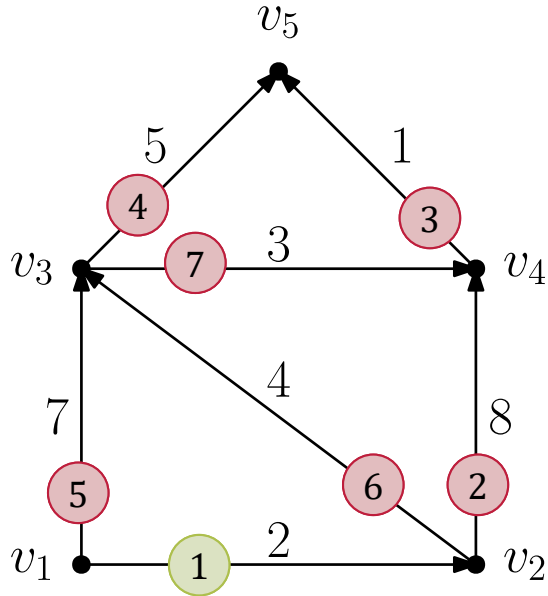
1. Function $MBF(D, c, s)$
2. $\ell(s) := 0$
3. $\ell(v) := \infty, p(v) := nil \quad \forall v \in V \setminus \{s\}$
4. For $i = 1$ to $n - 1$ do
5. For all $(v, w) \in A$ do
6. If $\ell(w) > \ell(v) + c((v, w))$ then
7. $\ell(w) := \ell(v) + c((v, w))$
8. $p(w) := v$

Beispiel



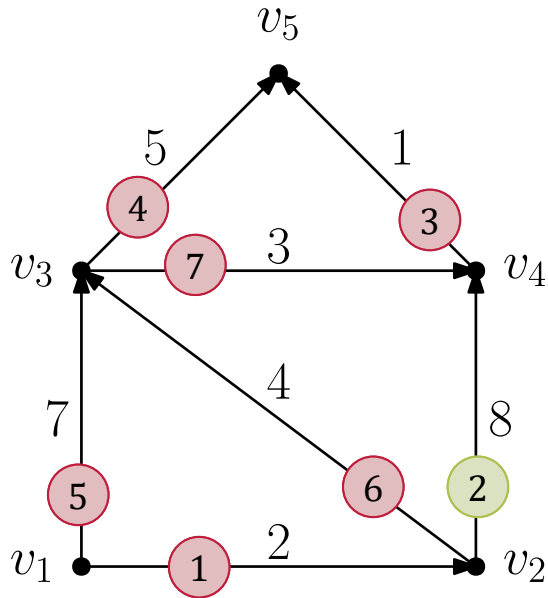
It.	v_1		v_2		v_3		v_4		v_5	
	$\ell(v_1)$	$p(v_1)$	$\ell(v_2)$	$p(v_2)$	$\ell(v_3)$	$p(v_3)$	$\ell(v_4)$	$p(v_4)$	$\ell(v_5)$	$p(v_5)$
0	0	—	∞	—	∞	—	∞	—	∞	—
1										
2										
3										
4										
5										

Beispiel



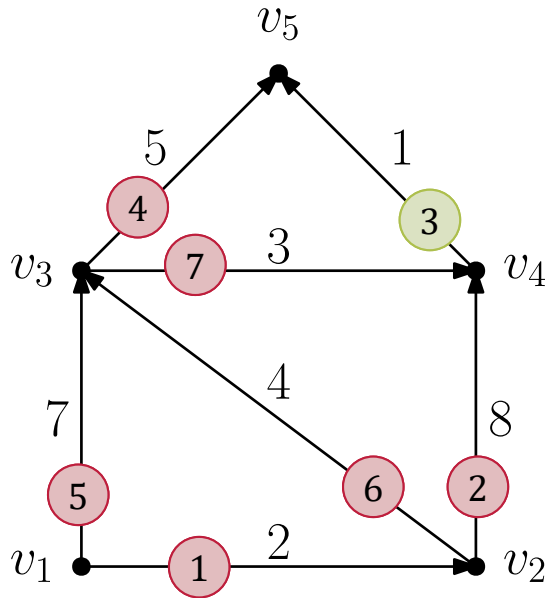
It.	v_1		v_2		v_3		v_4		v_5	
	$\ell(v_1)$	$p(v_1)$	$\ell(v_2)$	$p(v_2)$	$\ell(v_3)$	$p(v_3)$	$\ell(v_4)$	$p(v_4)$	$\ell(v_5)$	$p(v_5)$
0	0	—	∞	—	∞	—	∞	—	∞	—
1			2	v_1						
2										
3										
4										
5										

Beispiel



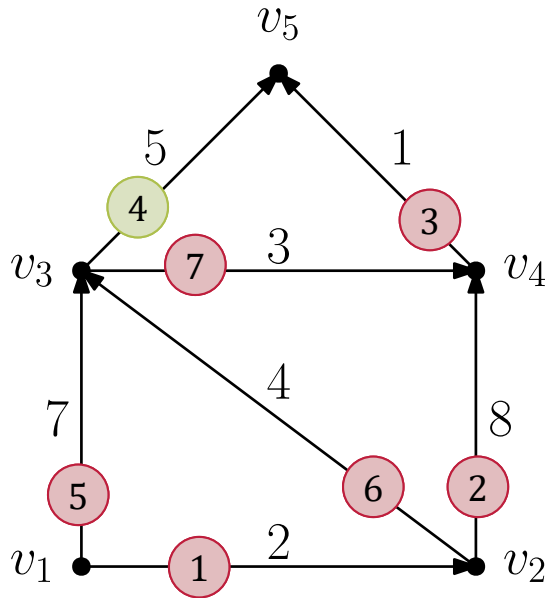
It.	v_1		v_2		v_3		v_4		v_5	
	$\ell(v_1)$	$p(v_1)$	$\ell(v_2)$	$p(v_2)$	$\ell(v_3)$	$p(v_3)$	$\ell(v_4)$	$p(v_4)$	$\ell(v_5)$	$p(v_5)$
0	0	—	∞	—	∞	—	∞	—	∞	—
1			2	v_1			10	v_2		
2										
3										
4										
5										

Beispiel



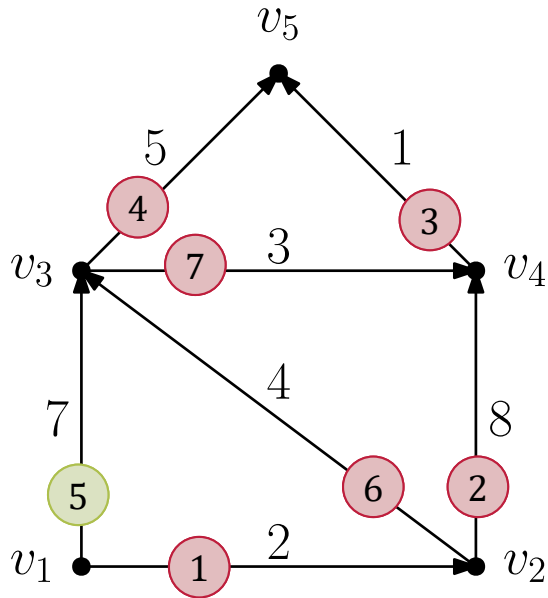
It.	v_1		v_2		v_3		v_4		v_5	
	$\ell(v_1)$	$p(v_1)$	$\ell(v_2)$	$p(v_2)$	$\ell(v_3)$	$p(v_3)$	$\ell(v_4)$	$p(v_4)$	$\ell(v_5)$	$p(v_5)$
0	0	—	∞	—	∞	—	∞	—	∞	—
1			2	v_1			10	v_2	11	v_4
2										
3										
4										
5										

Beispiel



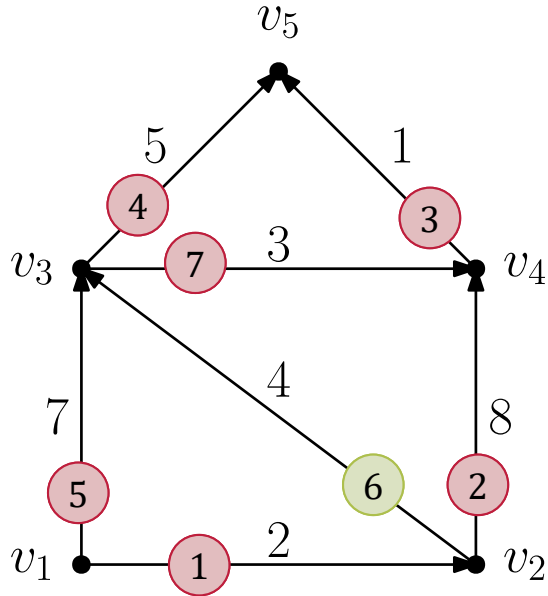
It.	v_1		v_2		v_3		v_4		v_5	
	$\ell(v_1)$	$p(v_1)$	$\ell(v_2)$	$p(v_2)$	$\ell(v_3)$	$p(v_3)$	$\ell(v_4)$	$p(v_4)$	$\ell(v_5)$	$p(v_5)$
0	0	—	∞	—	∞	—	∞	—	∞	—
1			2	v_1			10	v_2	11	v_4
2										
3										
4										
5										

Beispiel



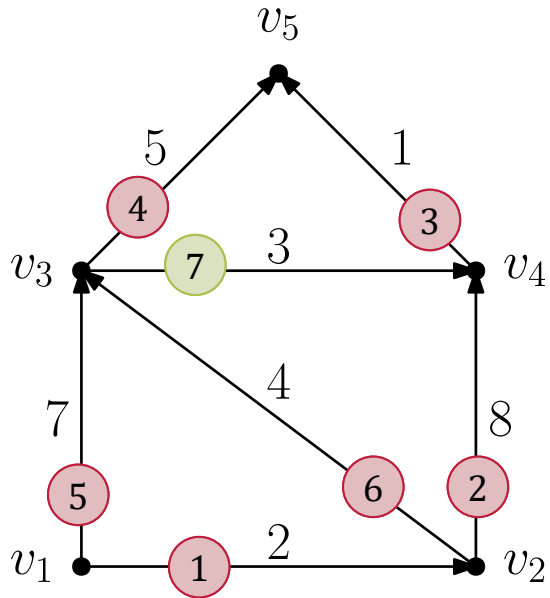
It.	v_1		v_2		v_3		v_4		v_5	
	$\ell(v_1)$	$p(v_1)$	$\ell(v_2)$	$p(v_2)$	$\ell(v_3)$	$p(v_3)$	$\ell(v_4)$	$p(v_4)$	$\ell(v_5)$	$p(v_5)$
0	0	—	∞	—	∞	—	∞	—	∞	—
1			2	v_1	7	v_1	10	v_2	11	v_4
2										
3										
4										
5										

Beispiel



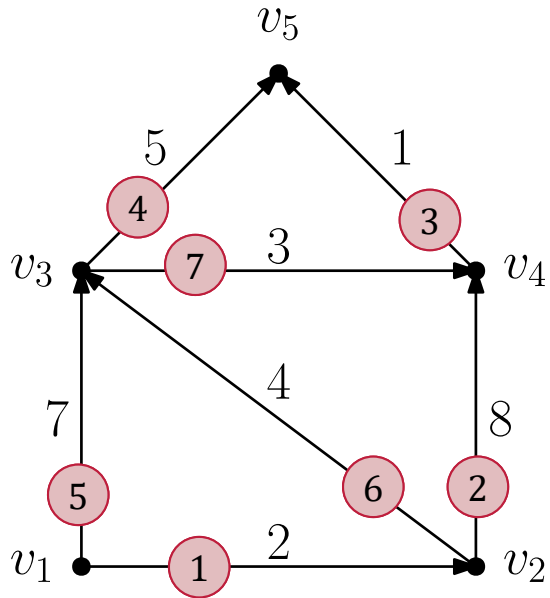
It.	v_1		v_2		v_3		v_4		v_5	
	$\ell(v_1)$	$p(v_1)$	$\ell(v_2)$	$p(v_2)$	$\ell(v_3)$	$p(v_3)$	$\ell(v_4)$	$p(v_4)$	$\ell(v_5)$	$p(v_5)$
0	0	—	∞	—	∞	—	∞	—	∞	—
1			2	v_1	6	v_2	10	v_2	11	v_4
2										
3										
4										
5										

Beispiel



It.	v_1		v_2		v_3		v_4		v_5	
	$\ell(v_1)$	$p(v_1)$	$\ell(v_2)$	$p(v_2)$	$\ell(v_3)$	$p(v_3)$	$\ell(v_4)$	$p(v_4)$	$\ell(v_5)$	$p(v_5)$
0	0	—	∞	—	∞	—	∞	—	∞	—
1			2	v_1	6	v_2	9	v_3	11	v_4
2										
3										
4										
5										

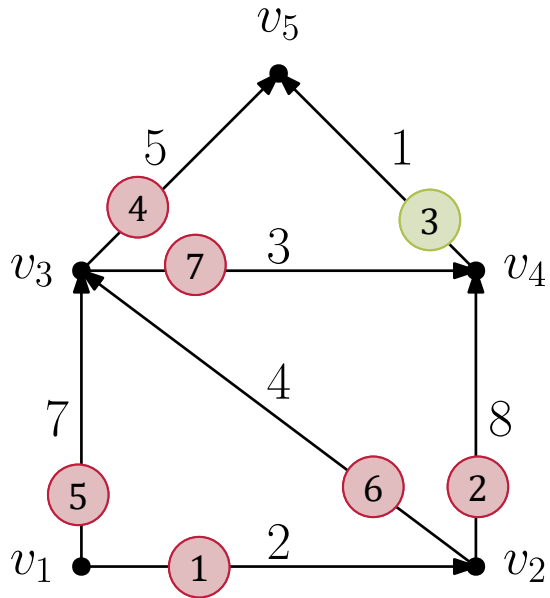
Beispiel



It.	v_1		v_2		v_3		v_4		v_5	
	$\ell(v_1)$	$p(v_1)$	$\ell(v_2)$	$p(v_2)$	$\ell(v_3)$	$p(v_3)$	$\ell(v_4)$	$p(v_4)$	$\ell(v_5)$	$p(v_5)$
0	0	—	∞	—	∞	—	∞	—	∞	—
1			2	v_1	6	v_2	9	v_3	11	v_4
2										
3										
4										
5										

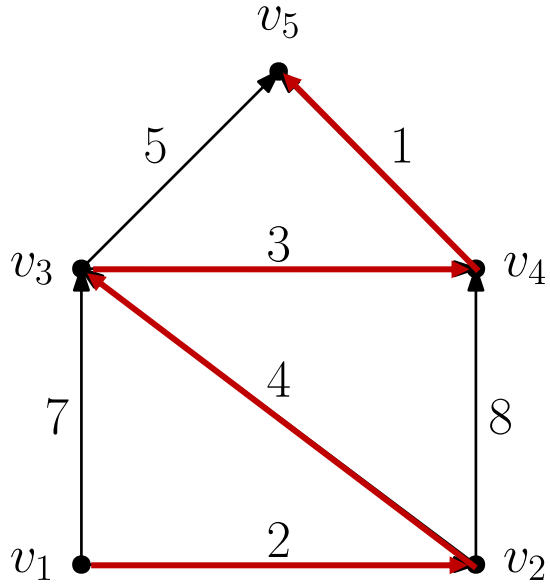
?

Beispiel



It.	v_1		v_2		v_3		v_4		v_5	
	$\ell(v_1)$	$p(v_1)$	$\ell(v_2)$	$p(v_2)$	$\ell(v_3)$	$p(v_3)$	$\ell(v_4)$	$p(v_4)$	$\ell(v_5)$	$p(v_5)$
0	0	—	∞	—	∞	—	∞	—	∞	—
1			2	v_1	6	v_2	9	v_3	11	v_4
2									10	v_4
3										
4										
5										

Beispiel



It.	v_1		v_2		v_3		v_4		v_5	
	$\ell(v_1)$	$p(v_1)$	$\ell(v_2)$	$p(v_2)$	$\ell(v_3)$	$p(v_3)$	$\ell(v_4)$	$p(v_4)$	$\ell(v_5)$	$p(v_5)$
0	0	—	∞	—	∞	—	∞	—	∞	—
1			2	v_1	6	v_2	9	v_3	11	v_4
2									10	v_4
3										
4										
5										

Algorithmus von MBF

Satz 3.10

Der Algorithmus von Moore, Bellman und Ford löst Problem 3.4 (Kürzeste Wege) korrekt für konservative Kantenkosten in $O(nm)$ Zeit.

Beweisidee:

Zeige in der i -ten Iteration:

1. Ist $\ell(x) < \infty$, dann existiert ein Pfad der Länge höchstens $\ell(x)$ in D
2. $\ell(x)$ ist höchstens die Länge eines kürzesten sx -Pfad in D mit maximal i Kanten

Allgemeine Kostenfunktionen

Lemma 3.11

Der Algorithmus von Moore, Bellman und Ford kann so erweitert werden, dass negative Kreise in einem Graphen $D = (V, A)$ identifiziert werden können.

Algorithmus 3.12

Eingabe:

Digraph $D = (V, A)$, Kostenfunktion $c: A \rightarrow \mathbb{R}$

Ausgabe:

Boolscher Wert, ob c konservativ ist.

1. Function ISCONSERVATIVE(D, c)
2. Wähle $s \in V$
3. Führe MBF(D, c, s) durch um Label $\ell(v)$ und $p(v)$ zu erhalten.
4. For all $(v, w) \in A$ do
5. If $\ell(w) > \ell(v) + c((v, w))$ then
6. Return true
7. Return false

Beweis Lemma 3.11

Zu zeigen:

D enthält negativen Kreis $\Leftrightarrow \ell(w) > \ell(v) + c((v, w))$ für einen Knoten $w \in V$ in Zeile 5.

„ \Rightarrow “:

Annahme: $\ell(w) \leq \ell(v) + c((v, w))$ für alle $w \in V$ in Zeile 5.

Betrachte Kreis C

Aufsummieren ergibt

$$\sum_{v \in V(C)} \ell(v) \leq \sum_{v \in V(C)} \ell(v) + \sum_{e \in E(C)} c(e)$$

Also

$$0 \leq \sum_{e \in E(C)} c(e)$$

„ \Leftarrow “:

Annahme: D enthält keinen negativen Kreis.

$\Rightarrow c$ ist konservativ

\Rightarrow kürzeste Pfade haben Länge $\leq n - 1$

\Rightarrow Keine „Verbesserung“ in der letzten Iteration

Pfingstwoche → Keine Vorlesung!

Aber Übermorgen gr. Übung!

+

Nächstes Mal: Quiz!