Karlsruher Institut für Technologie

Algorithmen für Routenplanung

12. Vorlesung, Sommersemester 2012

Daniel Delling | 11. Juni 2012

MICROSOFT RESEARCH SILICON VALLEY

Wiederholung Punkt-zu-Punkt

Anfrage:

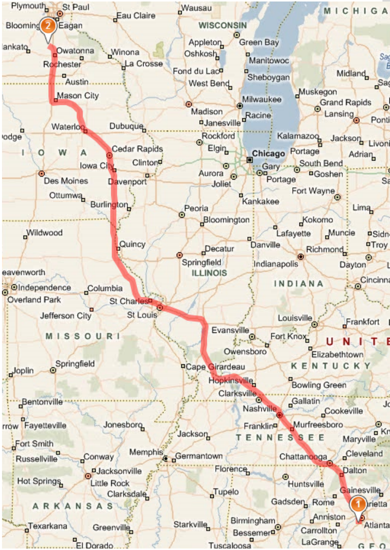
- finde die beste Route in einem Transportnetz

Idee:

- Netzwerk als Graph $G = (V, E)$
- Kantengewichte sind Reisezeiten
- kürzester Weg in G entspricht schnellster Verbindung

Ergebnisse:

- schnelle Algorithmen existieren



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Alternativ-Routen

Karlsruher Institut für Technologie

Anfrage:

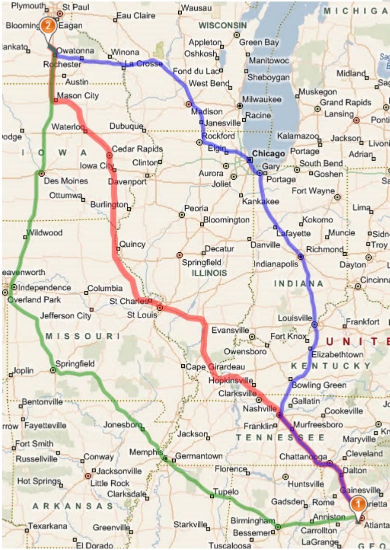
- finde gute Alternativen in einem Transportnetz

Problem:

- der kürzeste Weg ist wohl definiert
- Was aber ist eine gute Alternative?
- Problem erscheint rein heuristischer Natur
- nur auf den ersten Blick

Ziele:

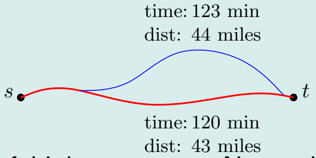
- lieber keine als schlechte Routen zeigen
- sollte nicht deutlich langsamer als Punkt-zu-Punkt sein



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Erste Ansätze

Optimiere andere Metriken

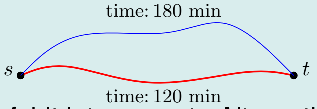


⇒ verfehlt interessante Alternativen

Berechne k-kürzeste Wege

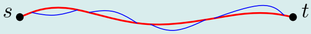
⇒ Alternative wahrscheinlich nicht unter den ersten 1000+ Pfaden

Disjunkte Pfade



⇒ verfehlt interessante Alternativen

++Gewicht auf kürzesten Weg



⇒ sieht seltsam aus

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Intuition

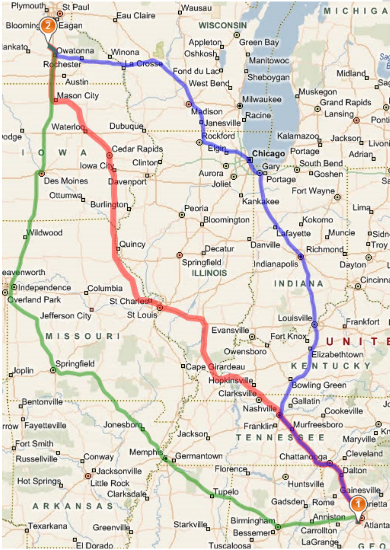
Alternativen sollten:

- nicht viel länger als der schnellste Weg sein
- signifikant verschieden

Erste Idee:

- finde einen Pfad, der Länge und Gemeinsamkeit minimiert
  - maixmal x% länger
  - teilt maximal y%

Ist das genug?



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Das dritte Kriterium

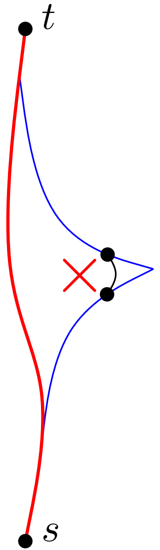
Problem:

- Routen können seltsam aussehen
- sogar wenn sie verschieden und kurz sind
- lokale Umwege
- User denkt sich “das kann ich besser”

Idee:

- kurze Subpfade müssen kürzeste Pfade sein
- beliebige Paare von Knoten auf dem Pfade sollen kleinen stretch haben

$\Rightarrow$ keine unnötigen lokalen Umwege



Eigenschaften eines Pfades P

- **Gemeinsamkeit:**
  der gemeinsame Teil von P und dem kürzestem Pfad *Opt*

$$\sigma(P) := \ell(Opt \cap P)$$

- **Lokale Optimalität:**
  für alle $u, v \in P$: wenn $\ell(P_{uv}) < lo(P)$, dann ist P_{uv} ein kürzester Pfad

$$lo(P) := \min_{u,v \in P} \{\ell(P_{uv}) \mid P_{uv} \text{ ist kein kürzester Pfad}\} + 1$$

- **uniformly bounded stretch:**
  der stretch zwischen beliebigen zwei Knoten auf P muss klein sein

$$ubs(P) := \max_{u,v \in P} \{\ell(P_{uv}) / \operatorname{dist}(u, v)\}$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

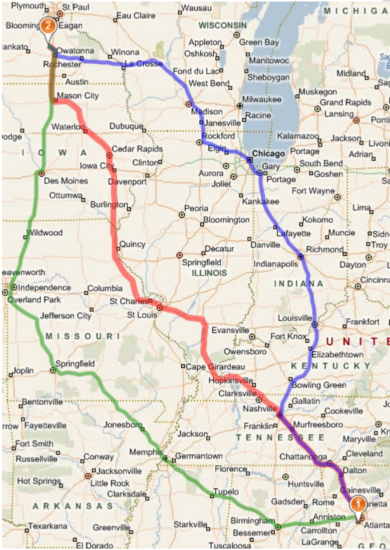
Gute Alternativen

ein gute Alternative *P*

- hat wenig Gemeinsamkeit mit dem kürzesten Weg *Opt*
- hat hohe lokale Optimalität
- hat niedrigen uniformly bounded stretch (UBS)

Idee:

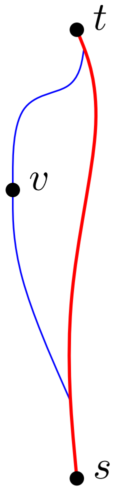
- definiere Optimierungsfunktion *f*
  - lineare Kombination der drei Maße
- finde *P* mit optimalem *f* und
  - *P* hat maximal *x*% Gemeinsamkiet mit *Opt*
  - *P* hat mindestens *y*% lokale Optimalität
  - *P* hat einen UBS von maximal *z*%



Einschränkung der Möglichkeiten

problems:

- hohe Anzahl von Pfaden
- effiziente Berechnung von lokaler Optimalität and UBS?
  - $|P|$ Dijkstra Suchen
  - $|P|^2$ p2p Anfragen
  - Many-to-Many?
  - $\Rightarrow$ zu langsam



Single via paths:

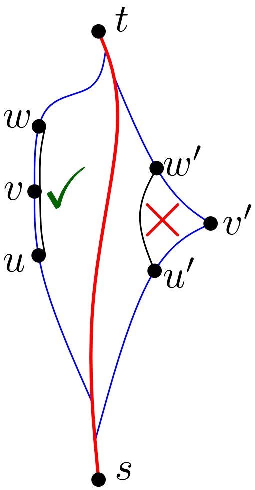
- Konkatenation von zwei kürzesten Pfaden: s–v und v–t
- Eigenschaften:
  - lineare Anzahl Pfade
  - P_v ist definiert durch via Knoten v
  - lokal optimal von s nach v und v nach t
  - müssen uns nur um den Bereich um v kümmern
  - wir können für UBS was zeigen
  - können effizient berechnet werden

Single Via Paths

2-Approximation von lokaler Optimalität:

- wähle zwei Knoten $u, w \in P_v$ die T vor und nach v sind
- checke ob der kürzeste Pfad von u nach w v enthält
  - **JA:** P_v ist T-lokal optimal
  - **NEIN:** P_v ist nicht $2T$-lokal optimal

$\Rightarrow$ schneller Test für lokale Optimalität



Uniformly Bounded Stretch

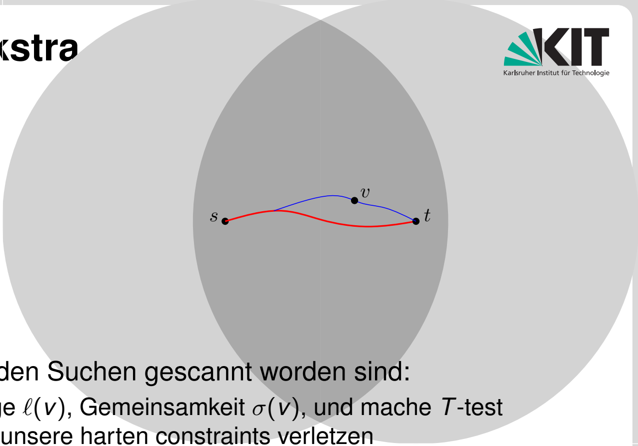
Lemma (Uniformly Bounded Stretch)

Wenn $\ell(P_v) = (1+\epsilon)\,\mathrm{dist}(s,t)$ und $lo(P_v) = \beta \cdot \mathrm{dist}(s,t)$ gilt, dann ist der UBS von P_v maximal $\beta/(\beta-\epsilon)$.

$\Rightarrow$ es reicht die Pfadlänge zu betrachten

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Erweiterter Dijkstra



Anfrage:

- starte 2 Dijkstras
  - von s
  - nach t
- stoppe Suche wenn radii größer $(1+\epsilon)\ell(Opt)$

finde Alternative:

- für alle v, die von beiden Suchen gescannt worden sind:
  - berechne Pfadlänge $\ell(v)$, Gemeinsamkeit $\sigma(v)$, und mache T-test
  - entferne alle v die unsere harten constraints verletzen
- gebe P_v, der Optimierungsfunktion f minimiert, aus

problem:

- Anzahl der Kandidaten ist sehr hoch ($\approx 1M$)

$\Rightarrow$ Anzahl der T-tests ist $\Rightarrow$ zu langsam

RE/CH

Beobachtung:

- unwichtige Teil werden geprunt
- die wichtigen Alternativen werden wohl nicht geprunt
- reduziert Anzahl der Kandidaten

Probleme:

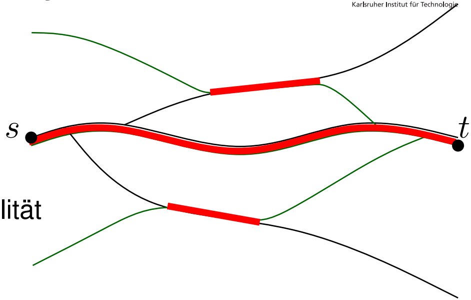
- Distanzen in den Suchbäumen ist inkorrekt
  - für jeden Kandidaten muss der Pfad durch 2 Queries rekonstruiert werden
- Anzahl Kandidaten immer noch zu hoch ($\approx 1K$)

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Bounding Local Optimality

Plateaus:

- Vorwärts- und Rückwärtsbäume schneiden sich
- definiere die Plateau-Länge $pl(v)$ für jeden Kandidaten v
- untere Schranke für lokale Optimalität
- Berechnung in linearer Zeit für alle Kandidaten möglich



Problem

- klappt nur wenn Bäume korrekt sind
- rein heuristisch für CH/RE
- aber korrekt für MLD

Algorithmen

X–BDV/ X–MLDV:

- bidirektionale BD/MLD-Query (mit erweitertem Stoppkriterium)
- sortiere Kandidaten nach $2 \cdot \ell(v) + \sigma(v) - pl(v)$
- gebe erste Knoten aus, der Nebenbedingungen erfüllt

X–REV/ X–CHV:

- bidirektionale RE/CH-Query (mit erweitertem Stoppkriterium)
- sortiere Kandidaten nach $2 \cdot \ell(v) + \sigma(v) - pl(v)$
- in dieser Sortierung:
  - rekonstruiere richtigen Pfad durch v
  - mache T-test
- gebe erste Knoten aus, der Nebenbedingungen erfüllt

$\Rightarrow$ nur 1–2 T-tests

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Experimente

input: Straßennetzwerk von Westeuropa

		PATH QUALITY							PERFORMANCE		
		success	UBS[%]		sharing[%]		loc opt[%]		#scanned	time	slow-
p	algo	rate[%]	avg	max	avg	max	avg	min	vertices	[ms]	down
1	X-BDV	94.5	9.4	35.8	47.2	79.9	73.1	30.3	16 963 507	26 352.0	6.0
	X-REV	91.3	9.9	41.8	46.9	79.9	71.8	30.7	16 111	20.4	5.6
	X-CHV	58.2	9.6	45.8	42.9	79.9	74.3	30.6	1 510	3.1	4.6
2	X-BDV	81.1	11.8	38.5	62.4	80.0	71.8	29.6	16 963 507	29 795.0	6.8
	X-REV	70.3	12.2	38.1	60.3	80.0	71.3	29.6	25 322	33.6	9.2
	X-CHV	28.6	10.8	45.4	55.3	79.6	77.6	30.3	1 685	3.6	5.3
3	X-BDV	61.6	13.2	41.2	68.9	80.0	68.7	30.6	16 963 507	33 443.0	7.7
	X-REV	43.0	12.8	41.2	66.6	80.0	74.9	33.3	30 736	42.6	11.7
	X-CHV	10.9	12.0	41.4	59.3	80.0	79.0	36.1	1 748	3.9	5.8

Erhöhen der Erfolgsrate bei Reach

Beobachtung:

- pruning entfernt eventuell gute Alternativen
- reach sagt aus, was der laengste kürzeste Pfad durch v ist

Idee:

- multipliziere alle reach-werte mit δ
- vergrößert den Suchraum

$\Rightarrow$ mehr Alternativen

$\Rightarrow$ langsamer

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Erhöhen der Erfolgsrate bei X–CHV

Beobachtung:

- sehr kleine Suchräume
- Anzahl Kandidaten sehr gering

Idee:

- vergrößer den Suchraum
- erlaube Abstieg in CH
- prune (u, v) nur, wenn Rank von v kleiner als Rank aller k Vorgänger von u

$\Rightarrow$ mehr Alternativen

$\Rightarrow$ langsamer

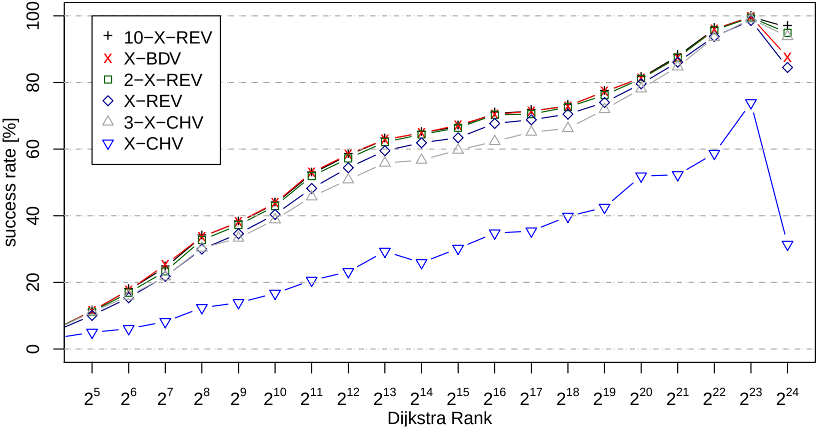
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Erhöhung der Erfolgsrate

		PATH QUALITY							PERFORMANCE		
		success	UBS[%]		sharing[%]		loc opt[%]		#scanned	time	slow-
algo	δ, k	rate[%]	avg	max	avg	max	avg	min	vertices	[ms]	down
X-REV	1	91.3	9.9	41.8	46.9	79.9	71.8	30.7	16 111	20.4	5.6
	2	94.2	9.7	31.6	46.6	79.9	71.3	27.6	31 263	34.3	9.4
	3	94.2	9.5	29.2	46.7	79.9	71.9	31.2	53 464	55.3	15.2
	4	94.3	9.5	29.3	46.7	79.9	71.8	31.2	80 593	83.2	22.8
	5	94.4	9.5	29.3	46.7	79.9	71.8	31.4	111 444	116.6	31.9
	10	94.6	9.5	30.2	46.8	79.9	71.7	31.4	289 965	344.3	94.3
X-CHV	0	58.2	9.6	45.8	42.9	79.9	74.3	30.6	1 510	3.1	4.6
	1	80.0	10.8	63.4	46.9	80.0	70.6	27.6	3 652	6.2	9.1
	2	87.5	11.5	52.5	45.8	80.0	67.8	26.2	6 756	9.4	13.9
	3	90.7	11.5	45.4	45.4	80.0	67.7	30.0	12 104	16.9	25.0
	5	92.9	11.3	41.9	44.8	80.0	66.8	27.9	39 835	55.2	81.8
	6	93.7	11.1	41.9	44.3	80.0	66.9	27.9	71 098	105.2	155.9
	8	94.3	11.0	41.9	44.0	80.0	66.8	27.9	210 046	368.8	546.4
	10	94.7	11.0	47.7	43.6	80.0	66.4	26.3	558 516	1 225.6	1815.7
X-BDV	–	94.5	9.4	35.8	47.2	79.9	73.1	30.3	16 963 507	26 352.0	6.0

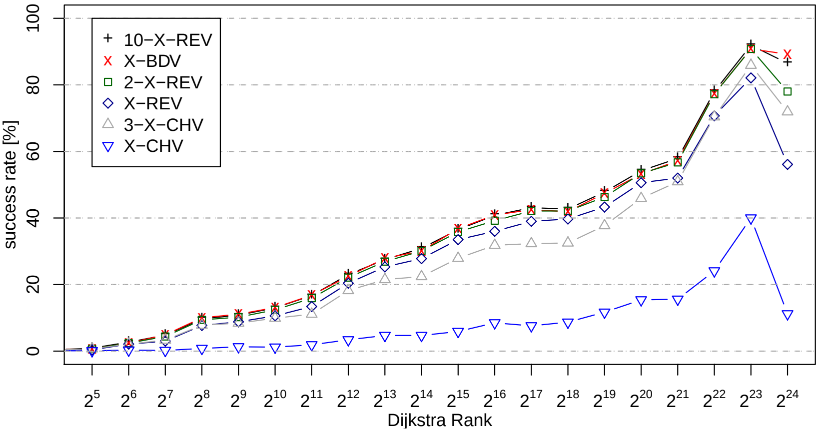
Lokale Queries

Erfolgsquote 1 Alternative



Lokale Queries

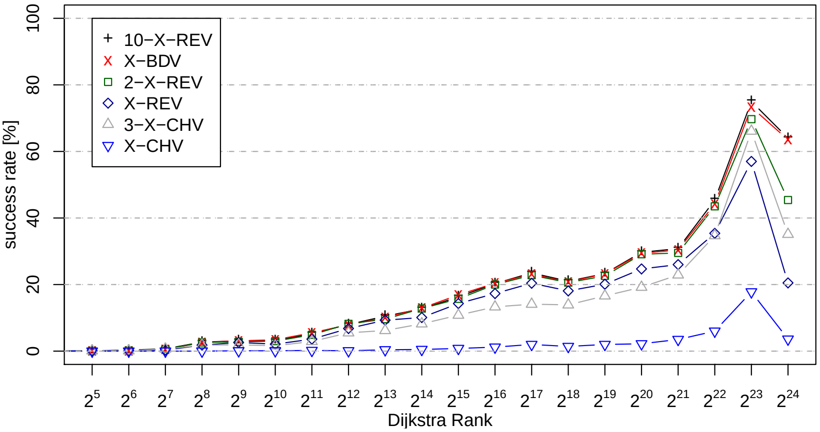
Erfolgsquote 2 Alternativen



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Lokale Queries

Erfolgsquote 3 Alternativen



Weitere Verbesserungen

Idee:

- benutze CHASE oder HL zur Rekonstruktion der Pfade (CH)
- Vorberechnung von guten Kandidaten
  - partitioniere den Graphen
  - bestimme für jedes Paar von Zellen gute Kandidaten
  - teste diese Kandidaten zuerst
  - benutze bisherige Algorithmen nur als fall-back

Ergebnisse:

- Berechnung von Alternativen nicht viel langsamer als Punkt-zu-Punkt Anfragen

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Zusammenfassung Alternativ-Routen

- Definition von Alternativ-Pfaden
- UBS, Gemeinsamkeit, Lokale Optimalität
- effiziente Algorithmen für single via Pfade
- nur 3–5 mal langsamer als point-to-point queries
- dieser Slow-Down kann noch weiter gesenkt werden
  - Vorberechnung von Kandidaten
  - schnellere Punkt-zu-Punkt Algorithmen

Offene Frage:

- Mit HubLabeling (und HLDB) vereinbar?

Literatur

Alternativ Routen:

- Ittai Abraham, Daniel Delling, Andrew V. Goldberg, Renato F. Werneck
  Alternative Routes in Road Networks
  In: *Proceedings of the 9th International Symposium on Experimental Algorithms (SEA '10)*, 2010
- Dennis Luxen, Dennis Schieferdecker
  Candidate Sets for Alternative Routes in Road Networks
  In: *Proceedings of the 11th International Symposium on Experimental Algorithms (SEA '12)*, 2012

Nächste Termine

Karlsruher Institut für Technologie

Mittwoch, 13.6.2012

Montag, 18.6.2012
Mittwoch, 20.6.2012
Montag, 25.6.2012