Efficient Route Planning

SS 2011

Lecture 12, Friday August 5th, 2011
(Course evaluation results, Transfer Patterns II)

Prof. Dr. Hannah Bast
Chair of Algorithms and Data Structures
Department of Computer Science
University of Freiburg

Overview of this lecture

- Organizational
  - Your results from Ex. Sheet #8 (Transfer Patterns)
  - Summary of your evaluation of the whole course
  - This is the **last** lecture
- Transfer patterns
  - Short recap
  - The direct-connection data structure
  - Feasible pre-computation using important stations
- Exam
- Current work at the chair

Feedback from ES#8 (Transfer Patterns)

- Summary / excerpts

  Stand 5.8 2:38

  - Aufgabe 1 (Evaluationsbogen) mit Hingabe ausgeführt
  - Gute Idee das zu belohnen
  - Ansonsten haben nur wenige die Aufgabe gemacht wegen keine Zeit und schon genug Punkte

Course evaluation results 1/5

- Contents of the course
  - Very interesting & relevant topic and algorithms (many)
  - Very practical, that's good (many)
  - Good balance between theory and practice (several)
  - Google Maps stuff was interesting (several)

Course evaluation results 2/5

- Style of the course
  - Competent and interesting explanations (many)

    ... but sometimes not in the first attempt (several)
  - Good, relaxed atmosphere (many)

    ... manchmal etwas zu viel "Späße" (one)
  - Much interaction with students (many)
  - Doing drawings / proofs "online" is instructive (several)
  - Implementation advice / live programming helps (several)

    ... "watching someone for 30 minutes trying to fix the code is a waste of everyone's time" (one)

Course evaluation results 3/5

- Exercises 1/2
  - Very interesting but also very time-consuming (many)

    ... but with the extra weeks it was ok (several)
  - Implementing very useful for understanding (many)

    "By implementing all the algorithms, you really learn something for lifetime"
  - Give more implementation advice earlier on (many)

    ... a lot of time spent in refactoring of old code (several)
  - Experimentation results were interesting / incentive (several)

    ... Experimentation results useless without clear params (one)

Course evaluation results 4/5

- Exercises 2/2
  - More feedback on the code would have been nice (several)

    "Hire more tutors which can actually help with the code" (one)

    "Offer a tutorial to discuss problems with the exercises. Sometimes questions are too complicated for a Forum" (one)
  - A lot of unnecessary code to write (one)

    "While it was stated that it was considered to be a valuable part of the learning process, it felt more like an excuse for not having prepared anything on the lecturers part."
  - "It felt like we are doing one algorithm after the other" (one)

Course evaluation results 5/5

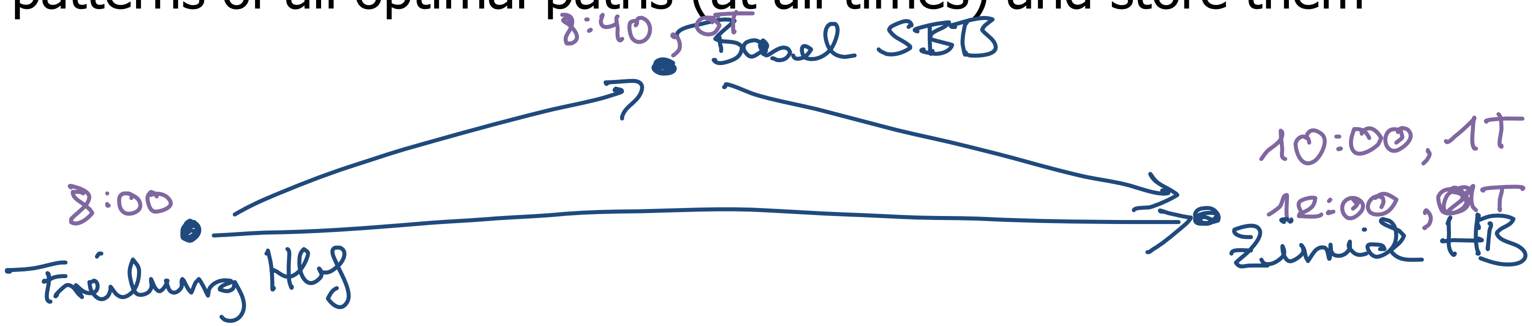
- Other
  - Video recordings were extremely helpful (many)
  - Java programmers had a big disadvantage (many)

    "Als Javanutzer fühlt man sich ein wenig wie der Depp"

    "Viele Implementierungsvorschläge gehen nur in C (hash map)"
  - The grade should depend on the exercise solutions (one)
  - Having an exam for this kind of lecture seems odd (one)
  - See other people's code after the deadline (one)

Transfer Patterns

- The basic idea on one slide
  - The **transfer pattern** of a path = the sequence of stations on the path where one boards, transfers, or alights
  - Idea: for each pair of stations, precompute all transfer patterns of all optimal paths (at all times) and store them



  - Then, at query time, do a time-dependent Dijkstra computation on this so-called **query graph**, where each arc evaluation is again a shortest path query, but restricted to **no transfers**
  - Such **direct-connection** queries are easy to compute fast

Components of a Transfer Pattern Router

- Transfer patterns precomputation
  - Compute (parts of) all transfer patterns of all optimal paths
- Direct-connection tables precomputation
  - Compute data structure for fast direct connection queries
- Query Graph Construction
  - Build the query graph of all transfer patterns between A and B
- Query Graph Evaluation
  - Dijkstra search on query graph, with arcs = direct connections
- Various Refinements / Optimizations
  - For example: filter out rare transfer patterns, …

Direct-Connection Queries

- One table per "line", let’s call this line L17

  Stations: S154 S97 S987 S111 …

  Time from start: 0min 7min 12min 21min …

  Start times: 8:15 9:15 10:15 11:20 12:20 …

- Lines per station … with positions in the respective line table

  Station S97: (L8, 4) (L17, 2) (L34, 5) (L87, 17) …

  Station S111: (L9, 1) (L13, 5) (L17, 4) (L55, 16) …

- Example query … from S97 @ 10:20 to S111
  - Intersect the lists of the two stations : (L17, 2 → 4) …
  - Find time from start to S97 and to S111 : 7min and 21min
  - Find first start time after 10:20 – 7min : 10:15 → **depart 10:22**
  - Compute arrival time at S111 : 10:15 + 21min → **arrive 10:36**

Important Stations 1/3

- The pre-computation so far is quadratic
  - Full Dijkstra to the whole graph for every station
  - Let m = #stations and n = #nodes
  - This amounts to a total of ~ m · n · L Dijkstra iterations

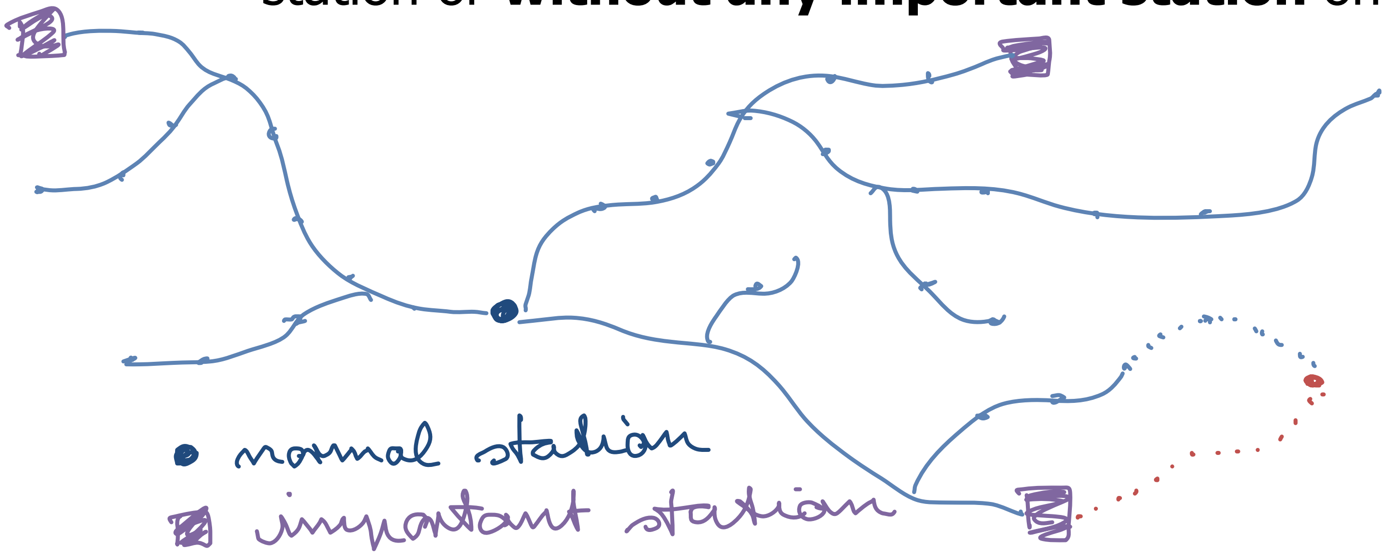
    where L is the average number of labels per node
  - A multi-label Dijkstra is ≈ 10 times slower per iteration than an ordinary Dijkstra (due to label set maintenance)
  - Example 1: m = 10K, n = 1M, L = 3, 10 µs / Dijkstra iter.

    30K seconds ≈ **80 hours**
  - Example 2: m = 1M, n = 1G, L = 3, 10 µs / Dijkstra iter.

    3G seconds ≈ **8 million hours ≈ 1000 years**

Important Stations 2/3

- How to improve on this?
  - Idea: Select 1% of all stations as "important"
  - Heuristic: where many paths transfer + geographic diversity
  - For each important station compute a global Dijkstra as before
  - For each non-important station, compute a local Dijkstra, that is, compute all **local paths** = all paths **until an important** station or **without any important station** on them



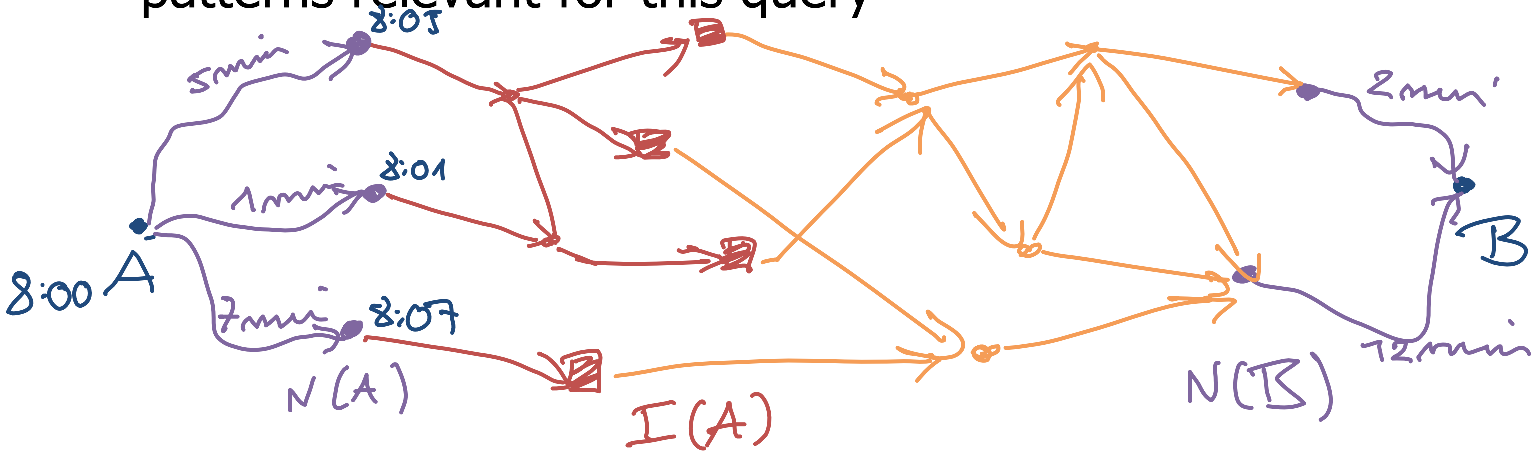
Important Stations 3/3

- Local Dijkstra search from a station s ... problem:
  - The number of (nodes on the) local paths is indeed small
  - But we have the usual "15 hours to the next village problem":

    If only one of the local paths has a large cost, say 15 hours, then the Dijkstra computation needs to search everything that can be reached from s within 15 hours
  - Unfortunately, almost every station has at least one local path of high cost, and hence our local Dijkstra searches end up being no less expensive than the global Dijkstra searches
  - Simple heuristic remedy: only consider local paths **up to two transfers**, that is, paths where more than two transfers are needed to get to an important station will be lost
  - Experience shows that these are **very rare** in practice

Query graph construction (sketch)

- For given source and target location A and B
  - Compute the sets N(A) and N(B) of stations near A and B
  - Get the precomp. local transfer patterns of these stations
  - Get the set~~s~~ I(A) ~~and I(B)~~ of important stations where the local paths from A ~~and from B~~ end
  - Get the global transfer patterns for each ~~pair of~~ important station~~s~~ ~~(~~a~~, b~~) where a ∈ I(A) ~~and b ∈ I(B~~)
  - Assemble this to form the query graph of all transfer patterns relevant for this query



Query graph search

- Time-dependent Dijkstra search
  - Start at the source location
  - For arcs from the source location to nearby station launch road network query (or have these precomputed)

    Same for arcs to the target location
  - For arcs between stations, ask direct-connection table

Exam 1/2

- The exam will be on Monday, August 15 at 2:00 pm
  - Here in HS 026 + it will last (only) 90 minutes
  - There will be **4 tasks**, out of which you can select **3 tasks**
  - Three **kinds** of tasks are possible
    - Execute an algorithm from the lecture, or some variant of it, on a given example (on paper)
    - Write a small program to solve a variant of a problem we have seen in the lecture
    - Compute, reason about, or prove a non-trivial (but also not very difficult) property of an algorithm or data structure from the lecture, or some variant of it
  - See the Search Engines WS 2009/2010 exam for examples

Exam 2/2

- The **bachelor students** (and only those)
  - ... must take an oral exam
  - On Wednesday, August 17, starting from 2:00 pm
  - In my office: building 51, 2nd floor, room 028
  - Questions will be of a similar kind as in the written exam, but of course not exactly the same

Work in my group

- Chair for Algorithms and Data Structures
  - Our work roughly subdivides as
    - 1/3 theory (new algorithms, complexity analysis, etc.)
    - 1/3 algorithm engineering (efficient implementations)
    - 1/3 software engineering (good, durable software)
  - Current projects
    - Route planning
    - Search engines, in particular: CompleteSearch & Broccoli
  - Current readings
    - http://ad.informatik.uni-freiburg.de/papers

References

- Transfer Patterns

  Fast Routing in Very Large Transportation Networks using Transfer Patterns

  Bast, Carlsson, Eigenwillig, Geisberger, Harrelson, Rachyev, Viger ESA 2010

  http://www.springerlink.com/content/c873271685124v42/

  http://ad.informatik.uni-freiburg.de/papers