Fast point-to-point shortest path queries on dynamic road networks with interval data

Giacomo Nannicini[1,2], Philippe Baptiste[1], Daniel Krob[1], Leo Liberti[1]

[1] *LIX, École Polytechnique, France*

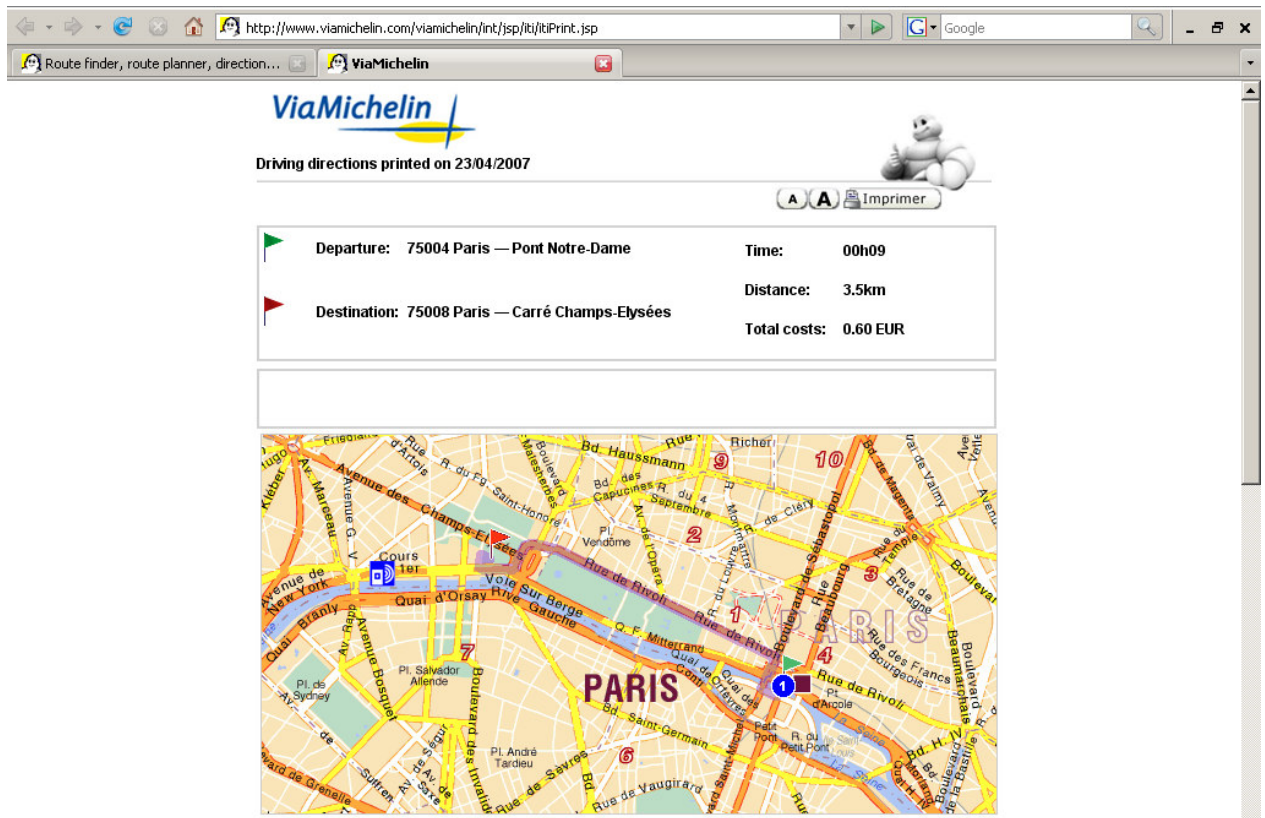
[2] *Mediamobile, Paris, France*

Summary of Talk

- Context Definition
- Problem Definition
- A Polynomial Time Approximation Scheme
- Computational Results
- Future Research

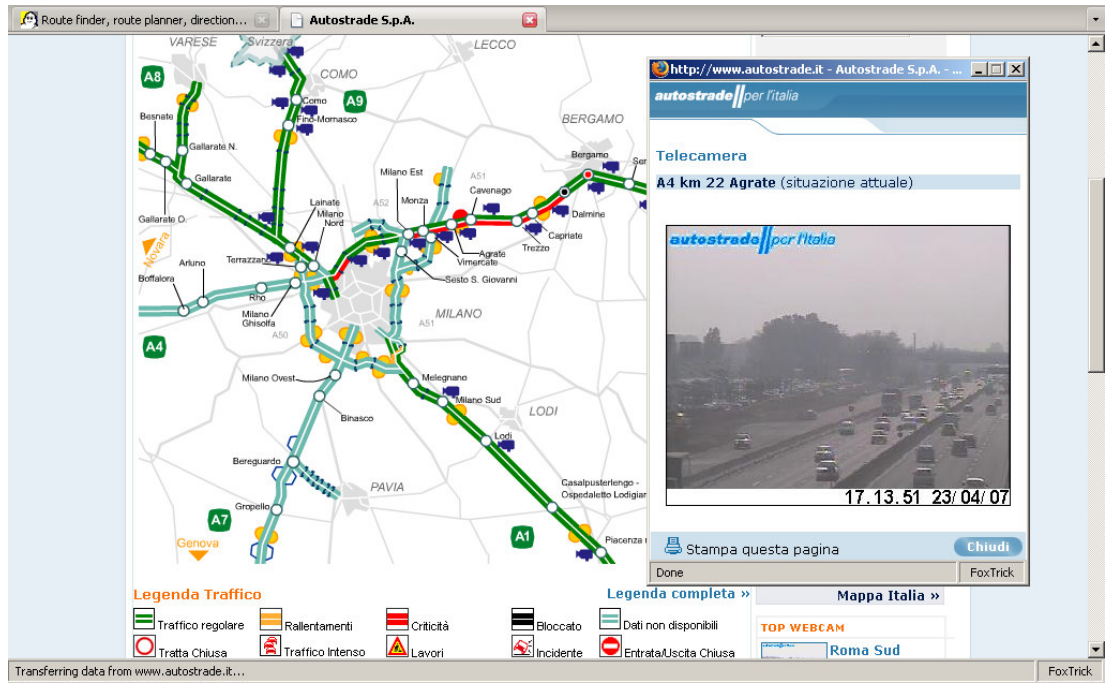
Context Definition

- Computing point-to-point shortest paths is of great interest to many users:
  - GPS devices with path computing capabilities
  - Many web sites provide users with route planners



Real-time Traffic Information

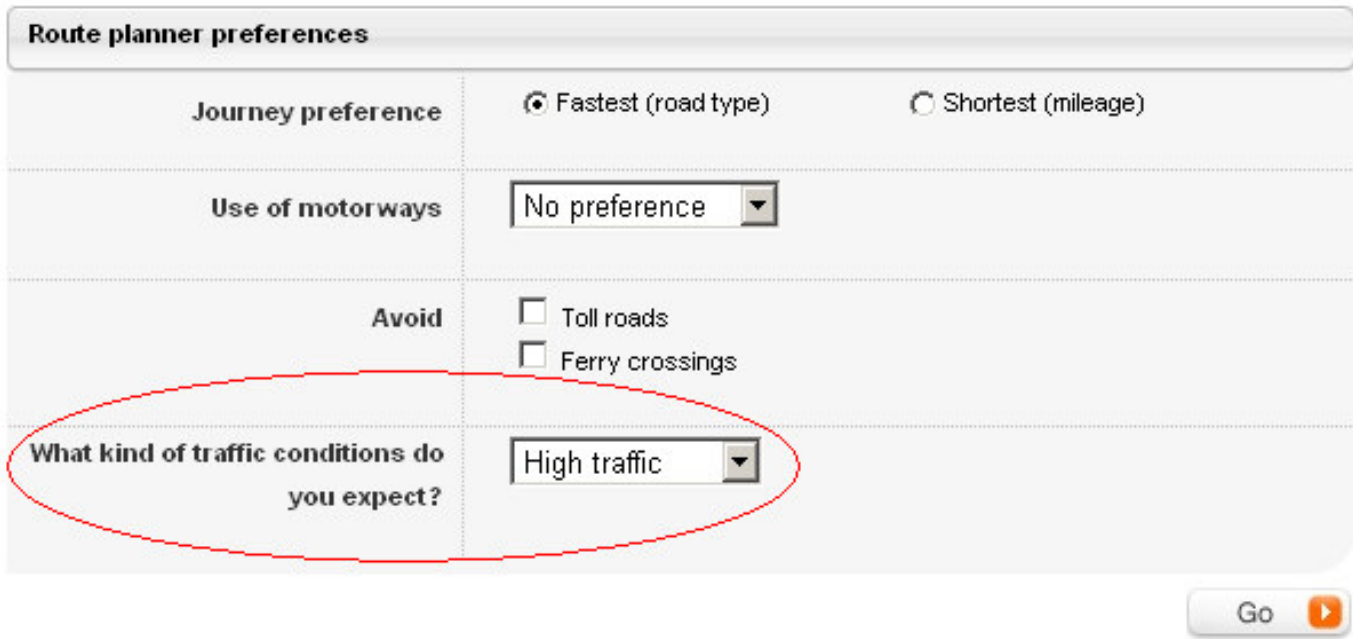
- As roads are being covered with various devices that allow traffic observation (cams, sensors, etc.), traffic information has become a frequent request from users



- Users are interested in the shortest path in terms of **travel time**, not of path's length

Traffic Prediction Tools

- Another common request from users is the computation of the fastest route with departure at a given time
- This means that we should **foresee** traffic conditions on each road, and then compute the shortest path
- It is a very difficult task!



Problem Definition (1)

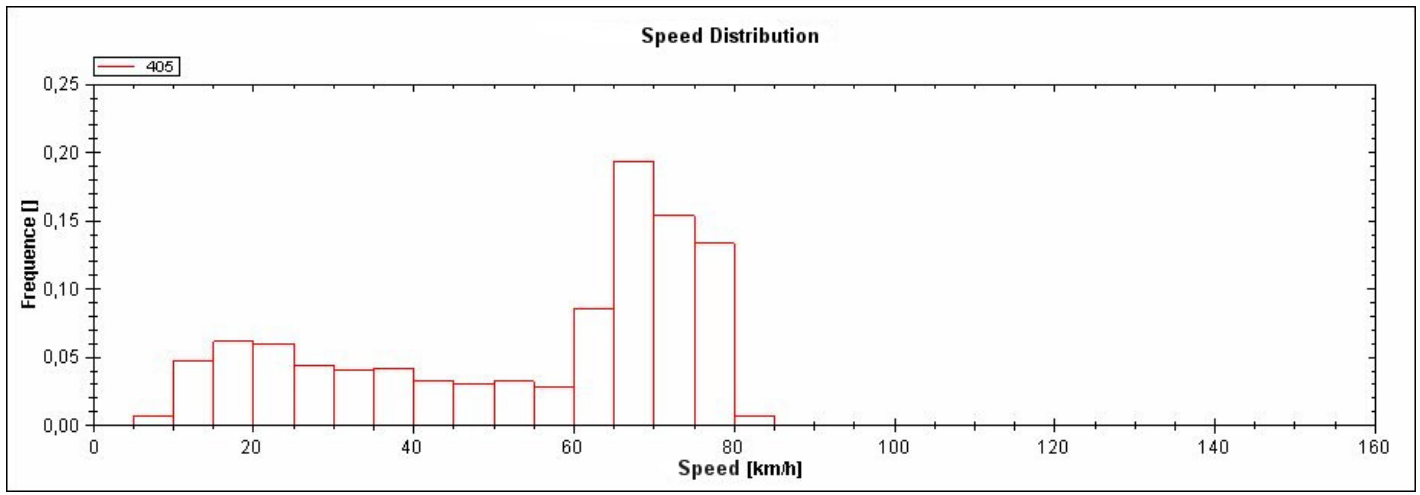
- We are given a graph $G = \langle V, A, c \rangle$ where $c : A \rightarrow \mathbb{R}$ is the cost function which associates a travel time to each arc, a source node s and a target node t; we want to compute the shortest $s \rightarrow t$ path (**P**oint **T**o **P**oint **S**hortest **P**ath **P**roblem)
- The cost function c changes over time
- If c is unbounded, we can find the optimum or a K-approximated solution $\forall K$ for the **PTPSPP** on a dynamic graph only with a labeling algorithm (e.g. Dijkstra's algorithm, A^*) which potentially explores the whole graph

Time Constraints

- The road network of a whole country can be very large
  - France: roughly 8M nodes, 17M arcs
- We want to be able to compute the shortest path in a low time
  - Dijkstra's algorithm takes on average 11 seconds: it's too much!
- There are several fast algorithms which compute exact solutions for the **PTPSPP** on a static graph (e.g. Highway Hierarchies, Reach + *ALT*), but they lose optimality whenever the cost function changes

A Reasonable Assumption

- Let us define lower and upper bounding functions l and u for c
  - This helps in simplifying our problem
  - Such functions can be computed using historical data on travel times
  - Historical data should be filtered to avoid outliers (e.g. to avoid $u = \infty$)



Problem Definition (2)

- Given a graph $G = \langle V, A, c \rangle$ with lower and upper bounding functions l and u such that $l(a) \leq c(a) \leq u(a) \, \forall a \in A$, for any pair of nodes $s, t \in V$ we want to compute the shortest $s \rightarrow t$ path for any cost function c.
- Query times should be as low as possible
- Since the cost function changes frequently, we can't perform costly update steps
- Pre-processing time (if reasonable) is not an issue

Notation

- For $s, t \in V$ let $P(s,t)$ be the set of all $s \to t$ paths
- Let $P_f^*(s,t)$ be the set of the shortest $s \to t$ paths on graph G with cost function $c = f$; e.g. $P_u^*(s,t)$ is the set of the shortest $s \to t$ paths on the graph weighted by the upper bounding function
- For $U \subset V$, $G[U]$ is the subgraph induced by U, and $P[U](s,t)$ is the set of $s \to t$ paths on graph $G[U]$
- $P_f^*[U](s,t)$ is the set of the shortest $s \to t$ paths on graph $G[U]$ with cost function $c = f$

Guarantee Regions (1)

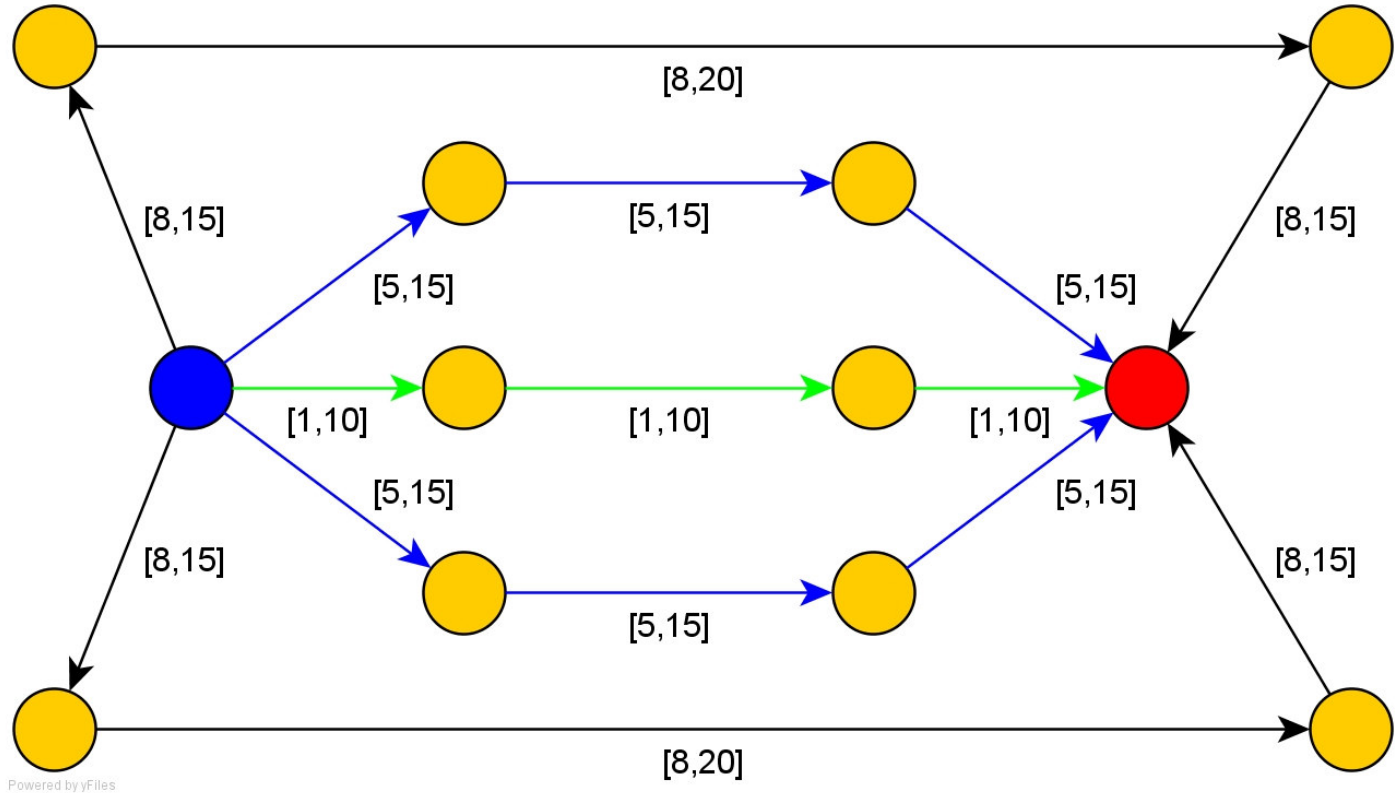
- For $K > 1$, $s, t \in V$ and path $p \in P_u^*(s,t)$, we define a guarantee region $\Gamma_{st}(K,p)$ as

$$\{v \in V | v \in p \vee \exists\, q \in P(s,t)\ (v \in q \wedge l(q) < \frac{1}{K} u(p))\}$$

- The following approximation property holds: for $p^* \in P^*(s,t)$ and $q^* \in P^*[\Gamma_{st}(K,p)](s,t)$, we have

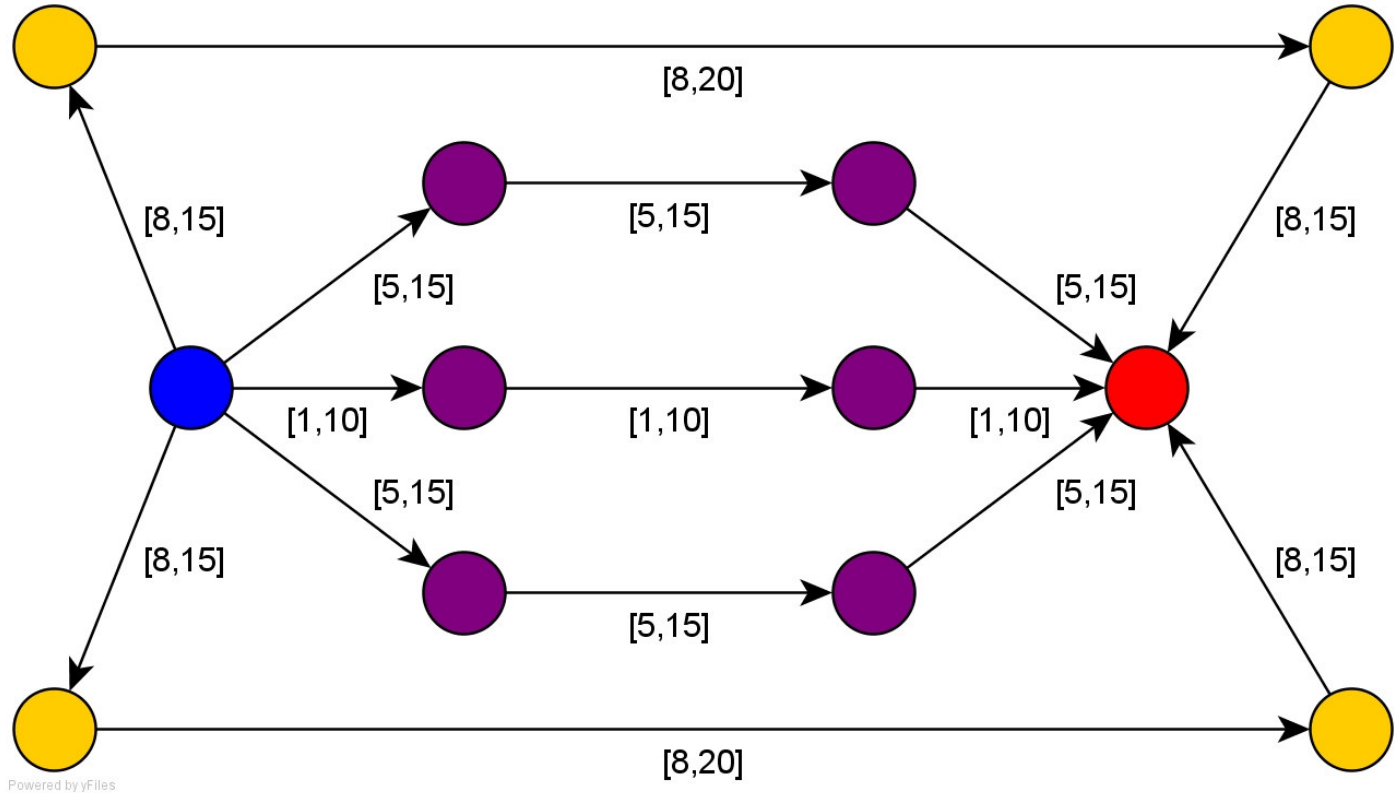
$$c(q^*) \leq K\, c(p^*)$$

Example (1)



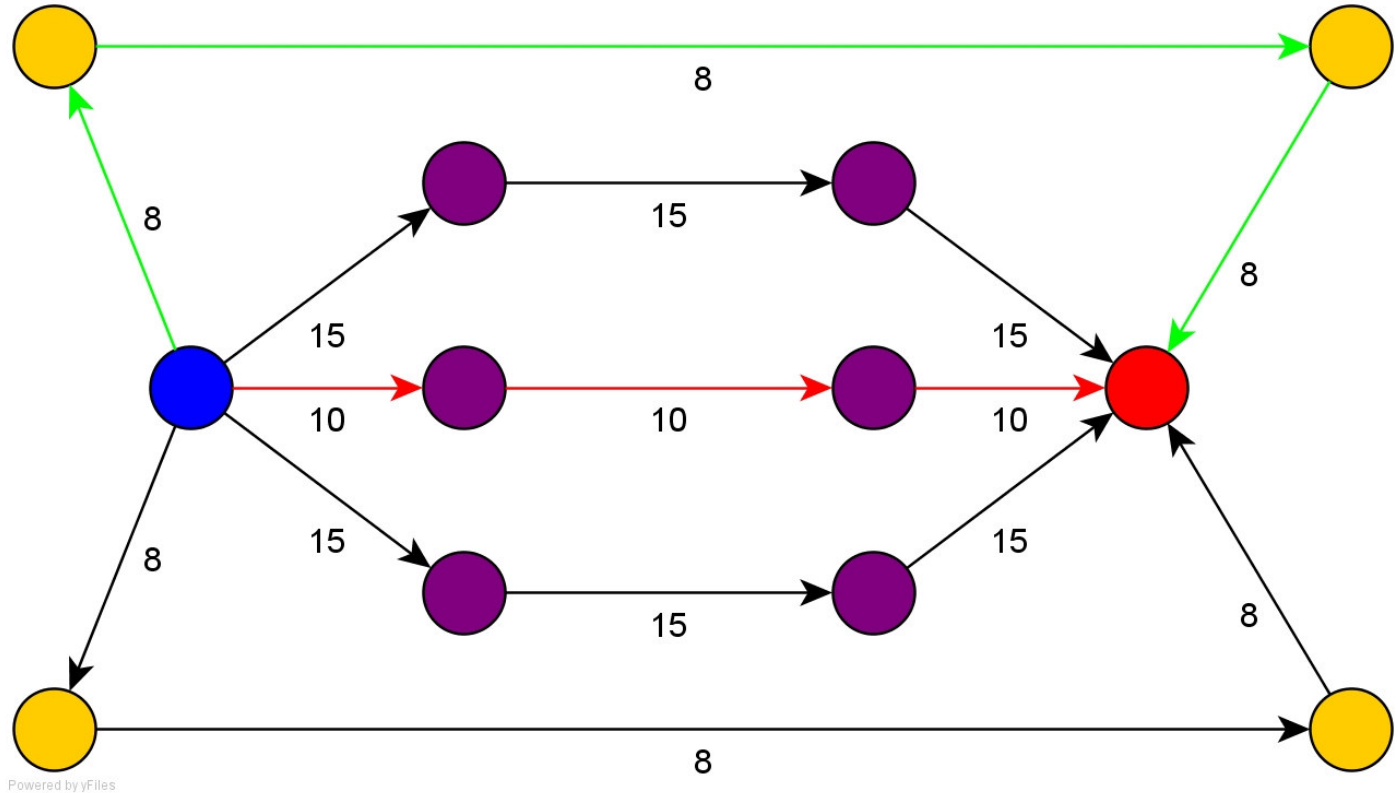
- Suppose $K = 1.5$
- Upper cost of green path: $u(p^*) = 30$
- Lower cost of blue paths: $l(p) = 15 \leq \frac{u(p^*)}{K}$

Example (2)



- $\Gamma_{st}(1.5, p^*) =$ blue, red and purple nodes.
- What happens in the worst possible scenario for this region?

Example (3)



- Shortest path has cost 24
- Shortest path restricted to $\Gamma_{st}(1.5, p^*)$ has cost 30
- $30 \leq 1.5 \cdot 24$, we are still within the approximation constant

Extending Guarantee Regions

- Computing $\Gamma_{st}(K, p)$ for all pairs $s, t \in V$ is not feasible
- Idea: compute a guarantee region for a set of source nodes, and a set of target nodes
- Define a covering of V with clusters $V_1, \ldots, V_k$ with the property that for all $i \leq k$ there are vertices $s_i, t_i \in V_i$ such that for all other vertices $v \in V_i$ there are paths $p \in P(v, s_i), q \in P(t_i, v)$ entirely contained in V_i.
- For all $i \leq k$ let $\sigma_i = \max_{v \in V_i, p \in P_u^*(v, s_i)} c(p)$ and $\tau_i = \max_{v \in V_i, p \in P_u^*(t_i, v)} c(p))$ be the costs of the longest shortest path in G_u from v to s_i and respectively from t_i to v over all $v \in V_i$.

Guarantee Regions (2)

- For $K > 1$, $i \neq j \leq k$ and a path $p \in P_u^*(s_i, t_j)$, we define the clustered guarantee region $\Gamma_{V_i V_j}(K, p)$ as

$$\{v \in V | v \in p \cup V_i \cup V_j \vee$$
$$\exists q \in P(s_i, t_j)\ (v \in q \wedge l(q) < \tfrac{1}{K}(u(p) + \sigma_i + \tau_j))\}$$

- The following approximation property holds: for $u \in V_i, v \in V_j$ (where $i \neq j$), $p^* \in P^*(u, v)$, $q^* \in P^*[\Gamma_{V_i V_j}(K, p)](u, v)$, we have

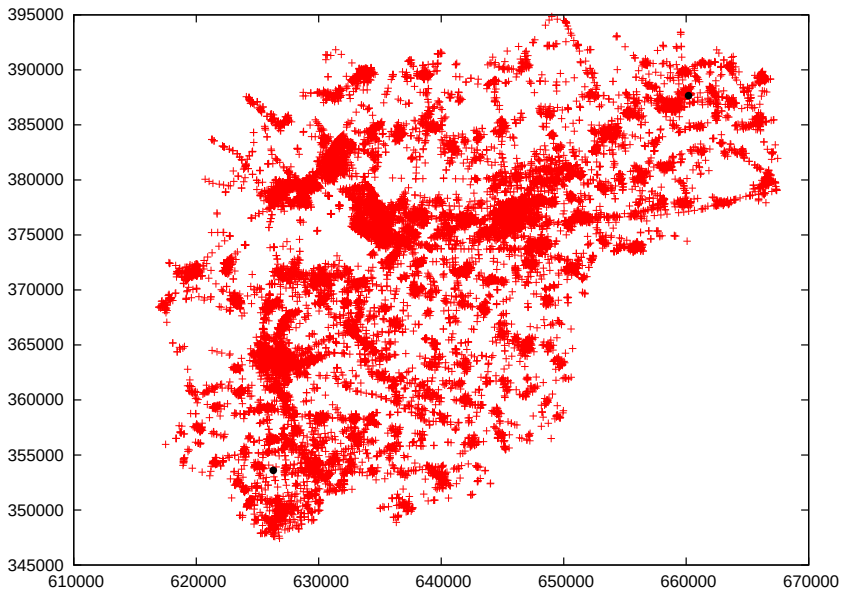
$$c(q^*) \leq K c(p^*)$$

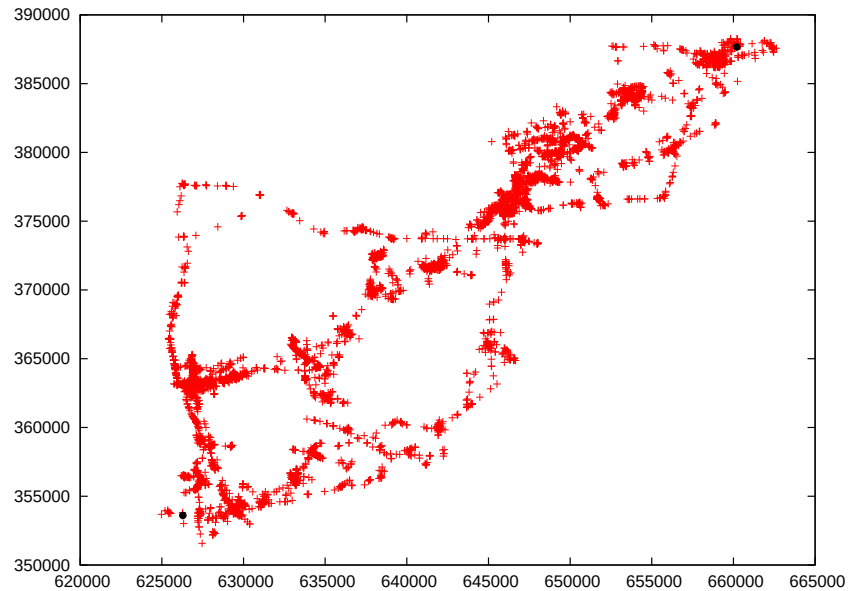
Framework

- Define a valid covering of V with clusters $V_1, \ldots, V_k$ with the aforementioned property
- Compute σ_i and τ_i for each cluster V_i
- Compute $\Gamma_{V_i V_j}(K, p)$ for each pair $i \neq j$ and for a given approximation constant K
- When computing the shortest path from s to t, apply Dijkstra's algorithm **only** on subgraph $G[\Gamma_{V_i V_j}(K, p)]$, where $s \in V_i, t \in V_j$
- The so-found path will have a cost within K times the optimum solution's cost
  - All computations can be done in polynomial time; it is thus a PTAS

Region's Cardinality

- Each region's size depends on l, u, the graph topology and the value of K
  - As K increases, the region's size decreases





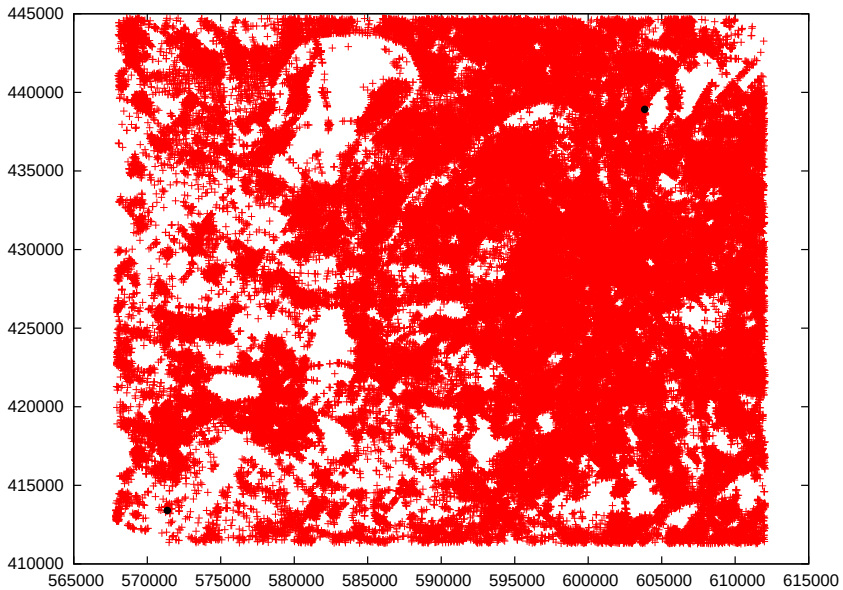
Effect on Performances

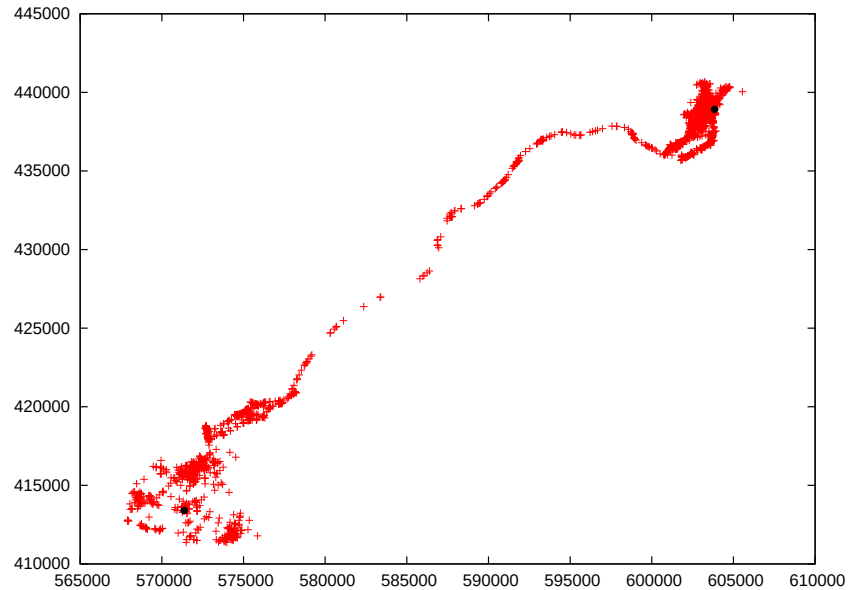
- The smaller the region, the less nodes we have to explore
  - This means faster query times, but of course the solution's quality can decrease
- A region's cardinality can also be used to give an upper bound to query times
- It is probably the most important decision to take, as it severely affects performances

Pros and Cons

- Pros:
  - Guarantee of a K-approximated solution
  - Query times can be reduced as much as needed (up to a certain degree!)
- Cons:
  - The speed-up is lower than that obtained with other techniques that pre-compute shortest paths
  - If K is not well chosen, we can have undesired behaviours

Extreme Cases





- Both are to be avoided!

Computational Results (1)

Unclustered graph (average values)

K	Search set size (Optimum sol.)	Search set size (Approx. sol.)	Cost increase	Speed up
3	74559	74532	0%	0%
4	74779	74219	0%	0%
5	74651	65126	0%	8.39%
6	74739	39282	0%	46.85%
7	74647	5609	0.07%	93.86%

Computational Results (2)

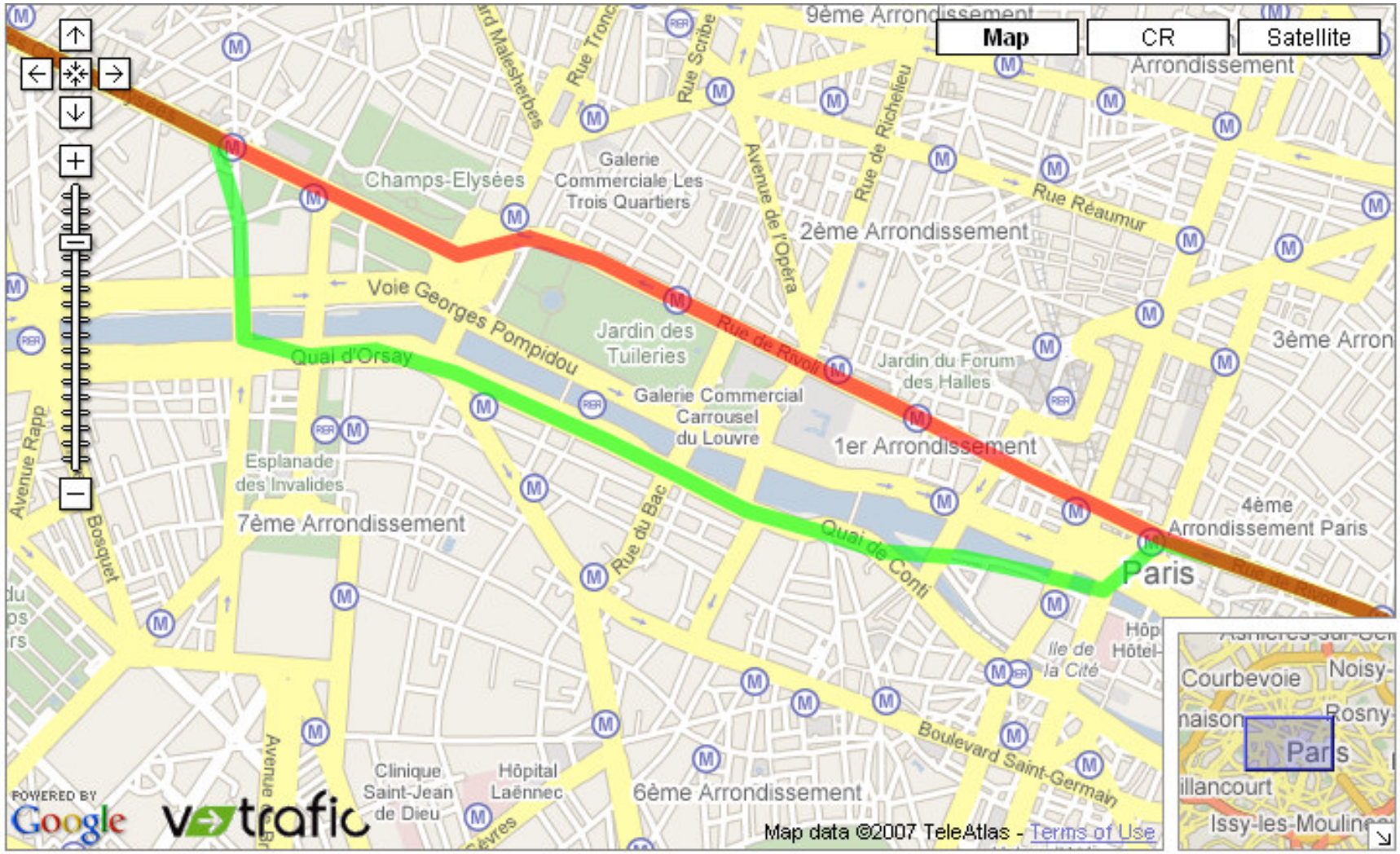
Clustered graph (average values)

K	Search set size (Optimum sol.)	Search set size (Approx. sol.)	Cost increase	Speed up
6	74493	73262	0%	0%
7	74605	66804	0%	5.83%
8	74129	56761	0%	20.35%
9	74436	34091	0.02%	54.26%
10	74494	13978	1.20%	82.05%

An Example

- Optimum solution: 5 minutes, 12 seconds
- Approximated solution: 5 minutes, 20 seconds

An example


Map
CR
Satellite
9ème Arrondissement
Champs-Elysées
Galerie Commerciale Les Trois Quartiers
2ème Arrondissement
Rue Réaumur
Voie Georges Pompidou
Quai d'Orsay
Jardin des Tuileries
Rue de Rivoli
Jardin du Forum des Halles
3ème Arron
Galerie Commercial Carrousel du Louvre
1er Arrondissement
Esplanade des Invalides
7ème Arrondissement
Rue du Bac
Quai de Conti
4ème Arrondissement Paris
Paris
Boulevard Saint-Germain
Clinique Saint-Jean de Dieu
Hôpital Laënnec
6ème Arrondissement
Courbevoie
Noisy-
Rosny-
Issy-les-Moulineaux
POWERED BY Google
Map data ©2007 TeleAtlas - Terms of Use

Future Research

- Computation of lower and upper bounds for all arcs
- Improvement in query times
- Efficient storage of node sets
- Faster pre-processing computations