Efficient Route-Planning Approach

With Limited Resources

Ondrej Moriš

xmoris@fi.muni.cz

Advised by

Assoc. Prof. RNDr. Petr Hliněný, Ph.D.

Faculty of Informatics
Masaryk University

GRAFY 2010

Motivation

What is it all about?

"Begin at the beginning and go on till you come to the end; then stop."
– The King, Alice's Adventures in Wonderland, Lewis Carroll

- **Route-Planning Problem**

  Given two positions in a road map, find the optimal path between them with respect to given optimality criteria.

  - be fast
  - be accurate
  - be reasonable
  - be memory efficient

  Example: find the fastest path from Belfast to Bear Haven.

Introduction

- road maps are represented by a specific class of graphs - *road map graphs*
  - almost planar, low average degree, very sparse, self-loops, parallel-edges, etc.
  - huge graphs - millions of vertices and edges
  - contain maneuvers – prohibited (forbidden) paths $M \in \mathcal{M}$
    - if a path does not contain any maneuver as its subpath, it is called $\mathcal{M}$-admissible

- route-planning $\Longleftrightarrow$ an instance of a well-known *SPSP problem*
  - Dijkstra's algorithm – $\mathcal{O}(|E(G)| + |V(G)| \log \log |V(G)|)$
    - in general good result, for road map graphs very slow and memory inefficient

- modern approaches use precomputed data to speed-up path searching
  - *preprocessing* – executed once in a while on a powerful computer
  - *queries* – executed on mobile device very often

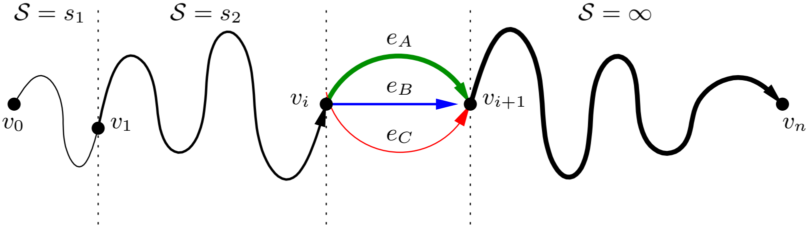
- heuristics are favored in practice – the exact paths usually involve issues

Our Approach

- two-level heuristic approach based on notions of a scope and a cell
  - Scope – a discrete mapping $\mathcal{S} : E_G \mapsto \{0, s_1, s_2, \infty\} \subseteq \mathbb{R}_0^+ \cup \{\infty\}$, $0 < s_1 < s_2 < \infty$.
    - assigned arbitrarily, e.g. according to road types
    - reflects importance of an edge in the global sense
    - must satisfy certain conditions to be applicable and efficient $G[E\restriction_{\mathcal{S}(e)\in\{\infty\}}(G)]$, $G[E\restriction_{\mathcal{S}(e)\in\{\infty,s_2\}}(G)]$, $G[E\restriction_{\mathcal{S}(e)\in\{\infty,s_2,s_1\}}(G)]$ strongly-connected
  - Cell – let (T', ν') be a partitioned branch-decomposition of a road map graph G, a *cell* is a subgraph $C \subseteq G$ such that $C = G[\nu'(l')]$, where l' is a leaf of T'.
    - *boundary* of a cell is a set of vertices that separates it from other cells, i.e. guts $\Gamma(e)$ of e, where e is the edge between l' and its parent
  - Boundary Graph – a highly reduced abstraction of a road map graph.
    - vertices – cell boundaries
    - edges – paths inside cells between boundary vertices using ∞ scope edges only

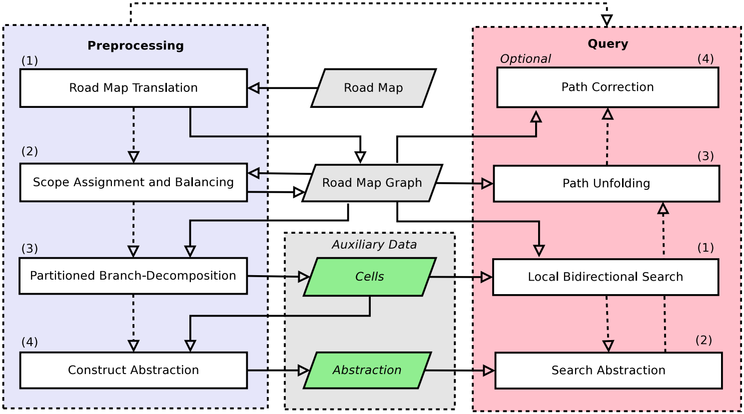
Our Approach

- An edge $e \in E(G)$ is *$\mathcal{S}$-admissible* for some path $P = s \cdots u \cdot e \cdot v \cdots t$ in a road map graph G if and only if the distance traveled from s to u or from v to t on edges of the scope higher than $\mathcal{S}(e)$ is less or equal than $\mathcal{S}(e)$. Path P is $\mathcal{S}$-admissible in G if all its edges are $\mathcal{S}$-admissible for P in G.

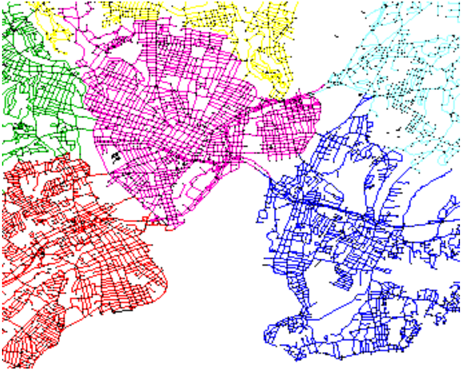


$$\frac{\textit{admissible}}{\mathcal{S}(e_A) = \infty}, \quad \frac{\textit{admissible}}{\mathcal{S}(e_B) = s_2}, \quad \frac{\textit{admissible} \Leftrightarrow |P_{v_1 v_i}| \le s_1 \vee |P_{v_{i+1} v_n}| \le s_1}{\mathcal{S}(e_C) = s_1}$$

Our Approach



Experimental Work: Preprocessing



Partitioned branch-decomposition respects natural road map disposition.
New Haven, CT, United States

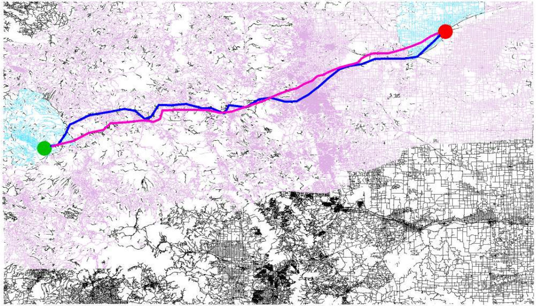
Experimental Work: Preprocessing

Preprocessing of 10 largest road maps in the United States.

Road Map G	$\|\mathbf{E(G)}\|$	$\|\mathbf{E(B_G)}\|$	**Reduction**	**Cells**	**Time**
Arizona	2 184 866	12 504	0.572%	420	25 min
Georgia	2 226 392	25 870	1.162%	400	14 min
New York	2 236 530	33 358	1.492%	422	47 min
Oklahoma	2 508 862	15 917	0.634%	436	23 min
Missouri	3 020 152	23 578	0.781%	519	34 min
Pennsylvania	3 081 096	29 411	0.955%	574	26 min
Virginia	3 333 864	33 854	1.016%	479	20 min
Florida	3 488 194	22 254	0.638%	655	30 min
California	5 394 762	16 002	0.297%	1 010	86 min
Texas	7 194 984	34 076	0.474%	1 336	135 min

(average cell size if 5 000 edges)

Experimental Work: Query



Search from the green source to the red target, the optimal $\mathcal{S}$-admissible path is blue, the optimal path is magenta. Vertices and edges visited during our search are light blue, during bidirectional Dijkstra search by light magenta.
Colorado, CO, United States.

Experimental Work: Query

Traditional comparison with Dijkstra's algorithm.

Road Map	Accuracy	Speed-up	Visited Vertices	Queue Size
Mississippi	98.3%	8	5623 / 185656	93 / 979
Idaho	98.1%	13	5159 / 290825	75 / 967
Kentucky	98.7%	8	6829 / 306304	325 / 1001
Tennessee	98.1%	10	5705 / 376629	97 / 983
Indiana	97.2%	10	5645 / 398078	145 / 1287
Minnesota	97.8%	6	5397 / 409200	124 / 1382
Georgia	97.7%	9	6293 / 530537	238 / 1549
Oklahoma	96.6%	8	6150 / 549373	123 / 1525
Ohio	98.1%	8	6413 / 577428	341 / 1560
Missouri	97.9%	8	6320 / 641253	123 / 1647
Pennsylvania	97.8%	8	6012 / 692769	205 / 1697
California	97.9%	9	6114 / 1147015	129 / 1973

(average cell size is 5 000 edges)

Conclusion

- new heuristic two-level route-planning approach has been developed
  - intended for computationally weak mobile devices and real-world road maps
  - based on the original notion of scope and well-known cell search technique
  - cells correspond to the road map graph partitioned branch-decomposition
  - provably optimal $\mathcal{SM}$-admissible path from the source to the target is found
- preprocessing
  - efficient, fast and scalable – approx. 2 hrs to preprocess Texas with 7M edges
  - auxiliary data of reasonable size – approx. 1% of the road map graph size
- queries
  - computing memory efficient – small queue size, less vertices are visited
  - accurary is good enough for practice – 98% at average
  - considerable speed-up – approx. 9 w.r.t Dijkstra's algorithm
- possible extensions: time-dependent route-planning, higher hierarchy
- future work: dynamic road maps, scope refinement

Thanks for your attention!