



Highway Hierarchies

一种最短路径搜索算法的构建与查询

一些基本的寻路算法

- 广度优先搜索 (BFS) : $O(|V| + |E|)$
- 经典的Dijkstra算法: $O(|V|^2)$
- 二叉堆的Dijkstra算法: $O((|V| + |E|)\log(|V|))$
- Fibonacci堆的Dijkstra算法: $O(|E| + |V|\log|V|)$
- A*算法: 复杂度与估价函数有关
- 各种双向优化的算法: 只能降低部分搜索的时间

整个西欧的公路网有1800万个节点，使用上述算法肯定是无法满足最短路径查找的需求！

Route Planning in Road Networks

- Highway Hierarchies (公路层次算法)
- Highway-Node Routing
- Many-to-Many Shortest Paths
- Transit-Node Routing



对于几百万顶点的交通图，只需要进行几十分钟的预处理，就能将每次最短路径的查询时间降到1s一下！



公路层次算法 (Highway Hierarchies)

- 算法的主要思想
- 名词定义
- 公路层次的构造
- 最短路径的查询

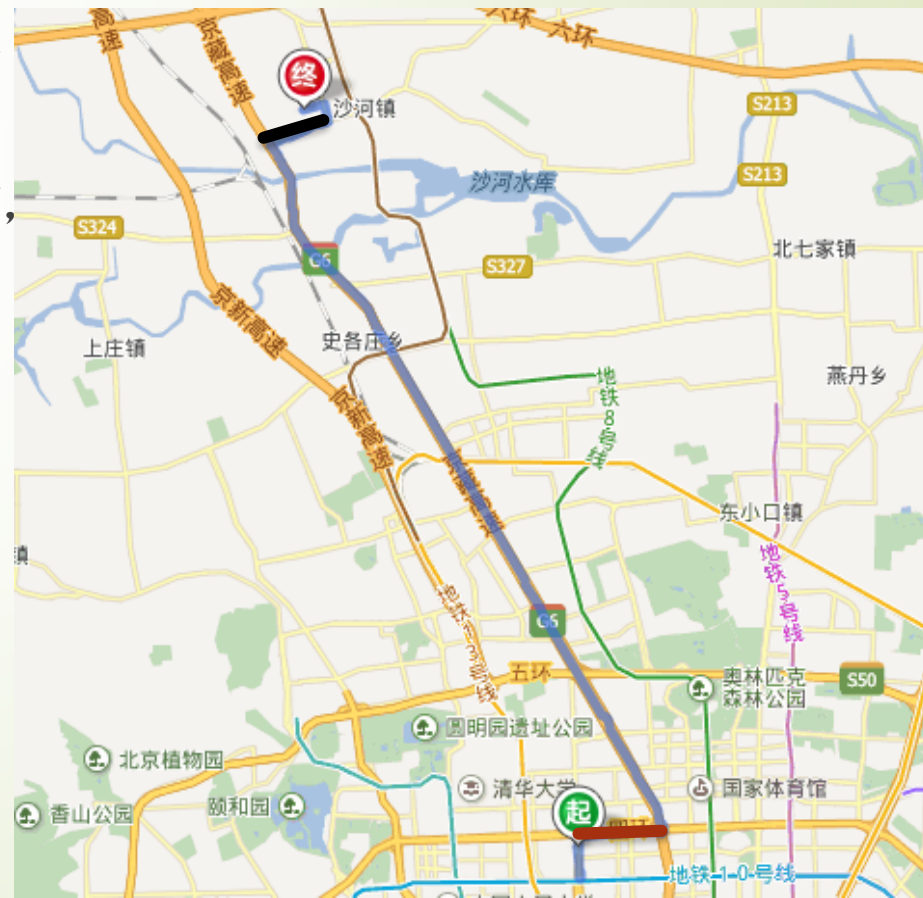
主要思想

我们在实际生活中寻路时，基本上遵循着这样一套简单的规则：

- 寻找一条可以通向目的地的高速公路，进入高速公路
- 在这条高速公路上行驶直至离目的点很近时，我们离开高速公路
- 寻找一条通向目的地的公路

比如右图：北航主校区→北航沙河校区：

- ▶ 学院路
- ▶ 京藏高速
- ▶ 百沙路



主要思想

一些商用的寻路算法的主要思想也是基于上述思想，即：

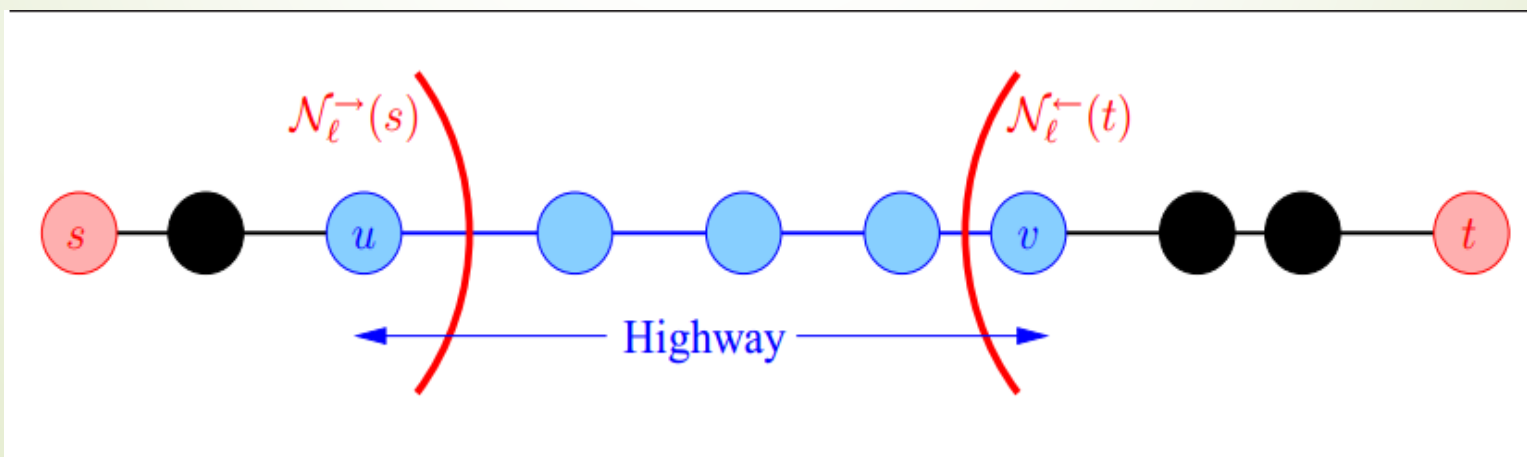
- 同时在源点和终点周围一定距离（比如20 km）中搜索所有路线
- 同时在源点和终点周围一定距离（比如100 km）中只搜索国道（notion roads）和高速公路（motorways）
- 再在离源点和终点周围更远的距离中只搜索高速公路

公路层次算法的主要思想和上述思想相差不大，也是将公路网分层再进行查询。

名词定义

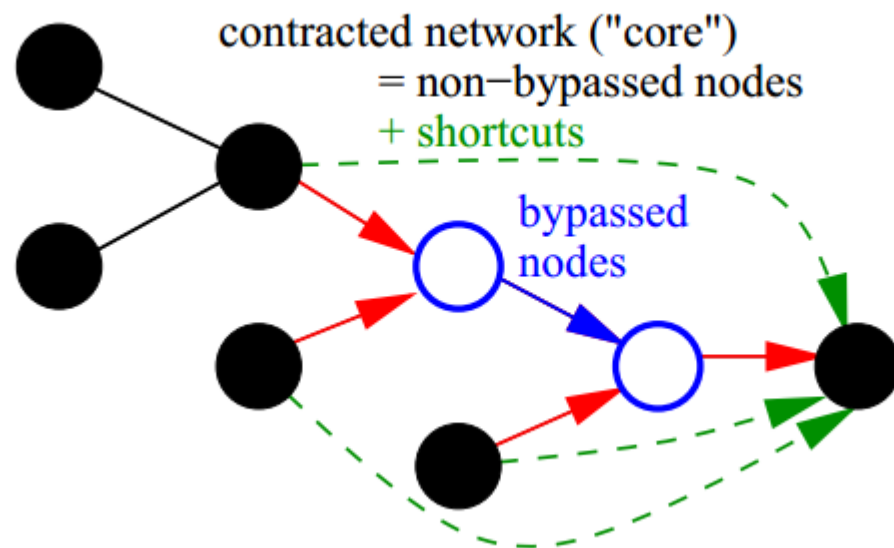
局部区域 (local area) : 对于公路网络中的每个顶点 u , 我们设定一个邻近半径 (neighborhood radius) H , 所有离 u 的最短距离不超过 H 的点称作 u 的邻居 (neighborhood)。通常来说 H 是一个可调参数, 对于不同的公路网络拥有不同的 H 值。

高速公路网 (highway network) : 对于网络中的一条边 $e=(u, v) \in E$, 如果 e 在图中某两个点 s, t 的最短路上, 即 s 到 t 的最短路包括 e , 且 u, v 不全是 s, t 的 neighborhood。那么我们称 e 在highway network中。



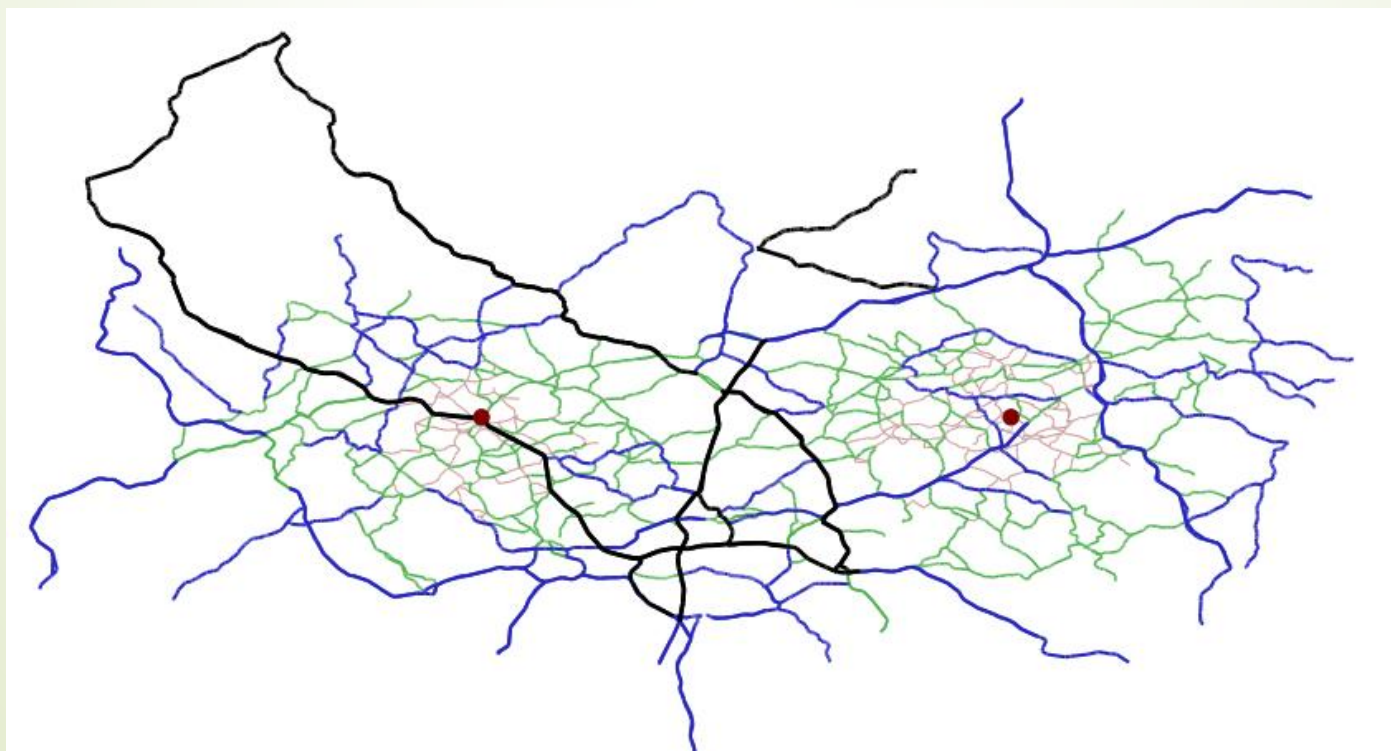
名词定义

收缩的高速公路 (**contracted highway network**)：在Highway中，包含了很多度很小的顶点，这些点 (*bypassed nodes*) 在寻路中没有任何作用，所以我们把它们删除，并添加上捷径 (*shortcuts*)，也就是将高速公路的多条边简化成一条边。我们把简化后的高速公路称作*contracted highway network*，也可以称作*core*。



名词定义

高速公路网络结构 (**highway hierarchy**) : core是可以递归计算的, 也就是说, 当我们计算出原公路网络的core后, 可以再此基础上再计算该网络的core.....我们把原本的公路网络称作第0层网络 (level 0), 从原网络中计算出的core称作第1层网络 (level 1) 等等。



公路层次的构造

对于一个图 G 来说, 它的highway hierarchy是 $G_1, G_2, G_3, \dots, G_l$, 我们可以通过下述方法求得各 G_i :