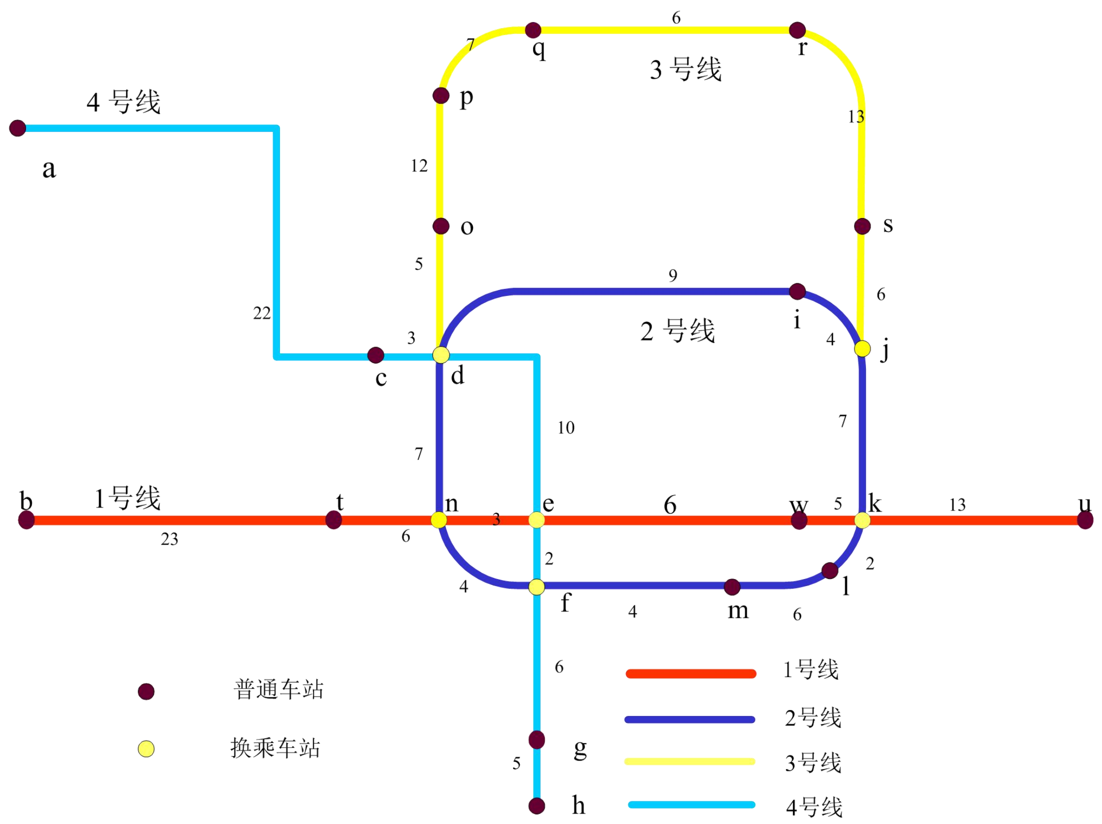
轨道交通图



MAP:

1,1,1,1,1,1,1,1,1

1,1,1,p,q,r,1,1,1

1,1,1,o,1,1,s,1,1

1,a,c,d,1,i, j,1,1

1,b,t,n,e,v,k,u,1

1,1,1,1,f,m,l,1,1

1,1,1,1,g,1,1,1,1

1,1,1,1,h,1,1,1,1

1,1,1,1,1,1,1,1,1

注：图中 w 改成 v 和代码一致

图为北京市城市轨道交通的部分网络，图中只列举了主要的车站和换乘站。线路上的数字表示的是在该区间的运行时分，下表为在换乘站换乘时的换乘时间。

<table>
<tr><th>换乘站</th><th colspan="2">换乘起、止线路</th><th>换乘走行时间（min）</th></tr>
<tr><td rowspan="6">d</td><td>4 号线</td><td>13 号线</td><td>8（14.25）</td></tr>
<tr><td>4 号线</td><td>2 号线</td><td>4（7.5）</td></tr>
<tr><td>13 号线</td><td>2 号线</td><td>9（15）</td></tr>
<tr><td>13 号线</td><td>4 号线</td><td>7（12.75）</td></tr>
<tr><td>2 号线</td><td>4 号线</td><td>4（8.25）</td></tr>
<tr><td>2 号线</td><td>13 号线</td><td>8（14.25）</td></tr>
<tr><td rowspan="2">f</td><td>4 号线</td><td>2 号线</td><td>4（7.5）</td></tr>
<tr><td>2 号线</td><td>4 号线</td><td>5（9.75）</td></tr>
<tr><td rowspan="2">j</td><td>13 号线</td><td>2 号线</td><td>5（9）</td></tr>
<tr><td>2 号线</td><td>13 号线</td><td>6（11.25）</td></tr>
<tr><td rowspan="2">n</td><td>2 号线</td><td>1 号线</td><td>4(7.5)</td></tr>
<tr><td>1 号线</td><td>2 号线</td><td>4(7.5)</td></tr>
<tr><td rowspan="2">e</td><td>4 号线</td><td>1 号线</td><td>3(6)</td></tr>
<tr><td>1 号线</td><td>4 号线</td><td>4(7.5)</td></tr>
<tr><td rowspan="2">k</td><td>1 号线</td><td>2 号线</td><td>5(9)</td></tr>
<tr><td>2 号线</td><td>1 号线</td><td>4(7.5)</td></tr>
</table>

求出 O（起点站）-D（终点站）的 K 短路（花费时间最少），就是从起点站到终到站的花费时间最少的路径、第二短路径、.......第 k 短路径。（限制条件是换乘不能超过三次，直到第 k 短路所花费的时间多于最小花费时间的 10min(包括等于 10min)时为止）

O-D 如下图（只需求对角线的右边部分）

	车站 a	车站 d	车站 g	车站 l	车站 j	车站 p	车站 r
车站 a							
车站 d							
车站 g							
车站 l							
车站 j							
车站 p							
车站 r							

即是车站 a 到达车站 d 的 k 短路：a-b-c-d

车站 a 到达车站 g 的 k 短路：

依次类推

算法

算法第一部分：计算网络上任意两点间的K条渐短路

Step1：初始化。给相关变量赋初值，建立空列表H用来存放每次迭代求得的最短路，空集合F用来存放各OD对及其对应的K短路；

Step2：取网络图上任意OD对$(v_i,v_j),v_i,v_j \in N$，设最大迭代次数NC；

Step3：迭代。利用蚁群算法求解(v_i,v_j)间最短路$d(v_i,v_j)$，在路径搜索过程中同时验证是否满足有效路径判断原则，若满足算法继续并将此次迭代得到的最后结果添加到列表H中，否则放弃此路径重新搜索。如果算法收敛或达到最大迭代次数，算法停止。在列表H中取出最短的K条渐短路，与OD对相对应存入F。此时这K条渐短路皆为有效路径，可将其作为下一阶段回溯验证的基础和标准；

Step4：重复Step2、Step3，直到所有的OD对均计算完毕。此时F中存在着所有OD对及其对应的K条渐短路。

算法第二部分：回溯验证寻找K短路

Step1：取OD对$(v_i,v_j),v_i,v_j \in N$，在OD对中v_i为起点，v_j为终点。初始化两个空集合$R^{(\alpha)},S$，$\alpha=0$，并将节点v_j放入集合$R^{(\alpha)}$中；

Step2：从集合$R^{(\alpha)}$中逐个取节点，设节点变量x_m，令x_m表示从集合$\mathrm{R}^{(\alpha)}$中取出的当前节点；

Step3：从x_m的第一个前驱节点开始，每个前驱节点都代表一条到达x_m的路径。设变量$l(x_i,x_j)$表示从x_j到x_i回溯过程中所经过的路段的长度和，初始化时令$l(x_i,x_i)=0$，$l(x_m^{(\eta)},x_j)=l(x_m^{(\eta)},x_m)+l(x_m,x_j)$，其中$x_m^{(\eta)}$表示$x_m$的第$\eta$个前驱节点，且$x_m^{(\eta)} \in N$。则在回溯过程中第$\eta$个前驱节点所在的路径的长度$d^{(\eta)}(v_i,v_j)=d(v_i,x_m^{(\eta)})+l(x_m^{(\eta)},v_j)$。检验该路径是否为有效路径，若$d^{r-1}(v_i,v_j)<d^{(\eta)}(v_i,v_j)<d^{r}(v_i,v_j)$，$r\in(2,3,...,k)$，则将$d^{(\eta)}(v_i,v_j)$插入到第r和第r+1个短路之间并将最后一个短路删除，同时将$x_m^{(\eta)}$存入集合S。否则，维持原K条短路不变。令$\eta=\eta+1$，重复Step3，直到η达到x_m的前驱节点的最大个数。

Step4：令$\alpha=\alpha+1$，$R^{(\alpha)}=S$，清空集合S。若$\mathrm{R}^{(\alpha)}$为空或仅有节点v_i，则

算法停止，现存的 K 条路径即为 OD 对 (v_i, v_j) 间的 K 条短路。

Step5：重复 Step2、Step3、Step4，直到所有的 OD 对均计算完毕。最终得到网络图中任意 OD 对间需参与客流分配的 K 短路。

配流

1. 已知现在所有 OD 之间的 K 短路和人数，求断面流量，所谓的断面流量就是一个小时内经过该路段的人数总和。

分两种情况：假如只有一条路径，就是所有的人都只有走这条路

加入有 K 条路径，就按照比例把人数分配到各条路径上，分配比例是这样计算的，

$$S(x) = e^{-2(C_k - C_{\min})^2/25} = e^{-2(5-3)^2/25}$$

$$P_k = \frac{S_k}{\sum S_i}$$

$C_{\min}$ 最小阻抗值，是指时间，

C_k -是 K 条路径的所用的时间，

假如，a-d 之间有 180 人，有两条 k 短路，分别是 a-b-d ，a-c-d 时间分别是 5,3 分钟，那么就按照比例进行计算：

a-b-d:

$$S(x)_{a-b-d} = e^{-2(C_k - C_{\min})^2/25} = e^{-2(5-3)^2/25} =$$

a-b-c: $S(x)_{a-c-d} = e^0 = 1$

分配比例是：

a-b-d: $P_k = \frac{S_k}{\sum S_i} = \frac{s(x)_{a-b-d}}{s(x)_{a-b-d} + s(x)a-c-d}$ 如此这样分配各个 od 之间的人数比例，

最后计算出 od 之间的断面流量，就是所有走过这段路径的人数之和。

The 1 K-path for OD: ad->a-c-d time:25 min(304)

The 1 K-path for OD: ae->a-c-d-e time:35 min(213)

那么经过 a-c、c-d、d-e 的人数分别是：304+213、304+213、213