**基于换乘次数最少的深度优先算法**

(1)换乘次数是乘客在选择换乘方案时的首要考虑条件，也是第一约束条

件。换乘次数通过被选择线路上的站点—线路交替序列个数可以表示出来。基于

三角形第三条边必然小于另两条边的理论，我们可以认为直达的换乘方案其行程

里程一定小于一次换乘方案的行程里程，以此类推。即使现实生活中存在直达方

案绕远路导致行程里程大于一次换乘方案的行程情况存在，基于从起始站点到终

到站点总体方向一定，即使绕路也不会太远的现实情况，乘客往往因不愿意换乘

而将其忽略了。

(2)行程时间最终也与换乘次数有关，因为行程时间多少取决于行程里程

长短，在此为了简化问题，我们只考虑公交之间的换乘，同时不计算换乘等车的

时间。那么行程时间就仅与每个方案所历站点总数成正比，则出行时间只与换乘

方案所历站数有关，站点总数则采用给同一线路上的站点排序的方法可以求到，

我们知道换乘次数越少行程就会越短，行程时间也就越短。

(3)乘车费用与换乘次数有关，因为城市公交通常都是固定票价，往往是

一到两元，但是每换一次车就会多买一次票，费用就会增加，由此可见乘车费用

多少也取决于换乘次数。

综上所述，任意站点之间最优换乘方案查询问题的多个约束条件可以简化为

一个，那就是最少换乘次数，这里还有三点需要说明：

1)站点距离问题——在实际中，公交站点的分布基本是均匀的，我们不需要考虑两个相邻站点间的距离问题，也就是在图1中连线的权值，不考虑权值可以使问题得到了简化。因为，在实际中公交站点的分布基本是均匀的，以致于人们甚至用从哪到哪几站路来衡量两地的距离。不考虑权值，也就相当于所有相邻站点间的距离都看做是 1，也就是我们常说的一站路，没有人去深究这一站路究竟是 300 米，还是 500 米。

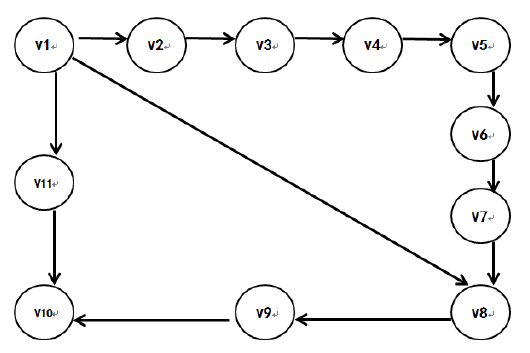


图1 公交线路图

2)最少换乘问题——最少换乘的要求使得程序在图的遍历时必须优先沿着线路前进，那么在选定一个线路进行遍历时就不用考虑其它，尽可能一条路走

到底。

3) 图的简化问题——如图2所示，图的顶点分两种，一种是线路经过

的点，一种是线路交汇的点。我们可以看出，单纯线路经过的点不会影响到换乘

方案，因为在经过这些点时除了选定的公交路线外没有与去其它站点的路线，所

以不会影响到换乘次数。

经过上面的分析，我们可以将图 1再次简化为图2，如下图所示。

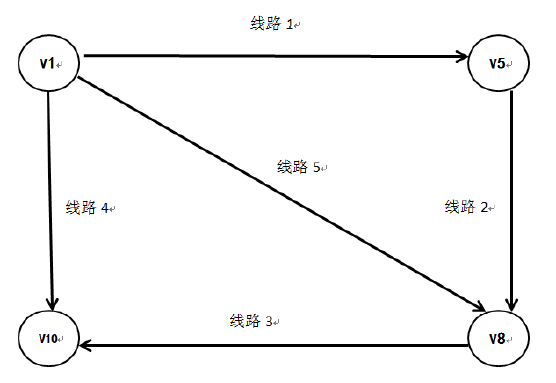


图2 简化公交线路图

总结来说，最少公交换乘问题在一般的图的问题上，给我们加了约束条件，使得我们很难用传统的迪杰斯特拉(Dijksrta)去解决任意站点间最优公交换乘问题，可同时增加的约束也使得公交换乘问题比在一般的图上找最短路径要简单的多，因为如上图所示，首先可以不用考虑图中各条边的权值，同时基于最少换乘这个约束条件，就必须尽可能的在一条公交线路上走到底，这使问题得到简化，我们认为可以采用普通的深度优先遍历搜索算法来分步解决它。

**基于换乘次数最少的深度优先算法**

公交系统中查找两个站点间的最优换乘方案与普通的图论中查找两点间的

最短路径是有区别，主要表现在：

(1)公交系统中图是由一条条公交线路组成的，每个线路包含一系列的站

点。

(2)公交系统中要求基于换乘次数最少为第一目标。 针对公交系统中这两条特殊点，决定了我们在查找两个站点间的通路时，必须优先沿着线路走，而且要先找直达的，然后找一次换乘的、再找两次换乘的，有需要的的话可以一直找下去，三次、四次、五次……。但在实际生活中，3 次以上的人们一般不再考虑了。

图2中任意两个站点间的线路段都可能包含其他的 0-n 个站点被忽略了，假设当我们要查找站点 V1到站点 V10的路径方案时，可以这么实现：

1. 查找直达的路径方案，将找到（V1—4—V10）。

2. 查找经过一次换乘的方案，将找到（V1—5—V8—3—V10）。

3. 查找经过两次换乘的方案，将找到（V1—1—V5—2—V8—3—V10）。

通过观察发现，查找两次换乘的方案可以通过将出发点从 V1 移动到 V5 转化为查找一次换乘方案。依次类推三次的也可以转化为两次，两次再转化为 1 次换乘。而且在查找的过程中，我们每选中一条线路就要沿线路，查找判断此线路上的所有站点，是否满足条件，可以看作是一种优化后的深度优先算法，它要求基于最少换乘次数、必须沿线路进行遍历。

**直达线路**

我们仍然以图2中，查找站点 V1到站点 V10的路径方案，首先查询经过站点 V1的公交线路得到线路集合 LList1，然后查找经过站点 V10的公交线路的到线路集合 Llist2，最后循环判断找出线路集合 LList1 和线路集合 Llist2 中所有相同的线路，相同的线路即满足同时经过站点 V1和站点 V10。在图 2 中线路 4 即经过了站点 V1又经过了站点 V10，则站点 V1到站点 V10的直达线路可以表示为(V1—4—V10)。

**一次换乘**

我们仍然以图2中，查找站点 V1到站点 V10的路径方案，首先查询经过站点 V1的公交线路得到线路集合 LList1，然后查找经过站点 V10的公交线路得到线路集合 Llist2，最后循环判断找出线路集合 LList1 中的线路与线路集合 Llist2中的线路之间的共用站点(除 V1、V10外)，共用站点即为换乘站点。在图中即经过站点 V1的线路5和经过站点 V10的线路 3 有共用站点 V8，则站点 V1到站点V10的一次换乘可以表示为(V1—5—V8—3—V10)。

**二次以上换乘**

仍然以图2中查找站点 V1到站点 V10的路径方案为例，首先查询经过站点V1的公交线路得到线路集合 LList1。然后循环基于深度优先在线路 LList1 (i)上面移动起始站点 V1，当 V1移动到 V5时，V5到V10采用前面的一次换乘，将找到(V5—2—V8—3—V10)，结 合 V1到 V5移 动， 二 次 换 乘 方 案(V1—1—V5—2—V8—3—V10)将被找到。两次以上采用递归的方法，支持找到更多的换乘，由于公交问题的现实情况，3 次以上不再考虑，所以递归到 3 次换乘将退出。