

数学建模算法与实践

最短路径问题建模

HY Deng

dhy0826@126.com

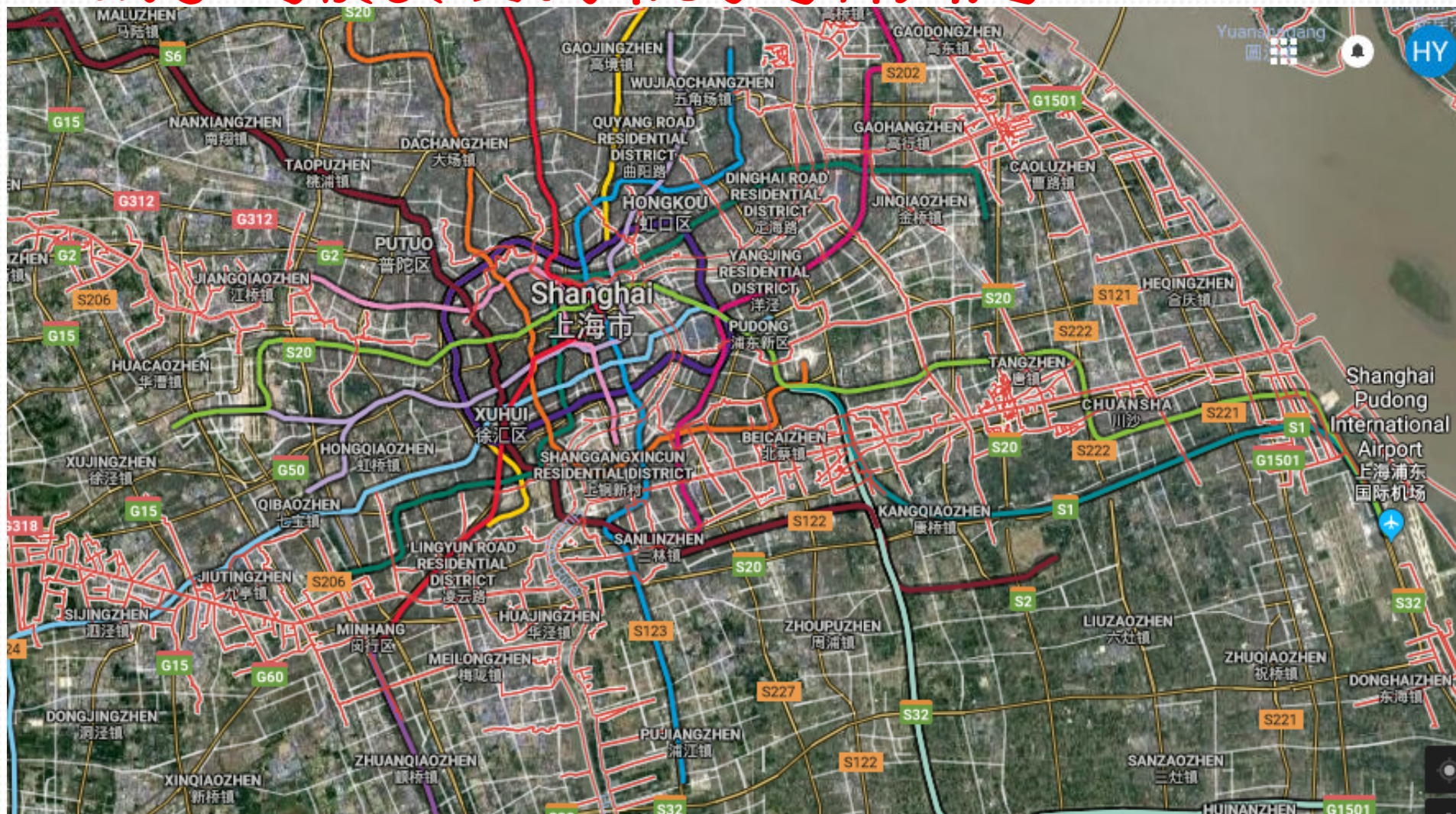


内容提要

- ① 引例：最短路径算法介绍
- ② 逢山开路问题
- ③ 交巡警平台设置与调度问题
- ④ 公交线路选择问题（见交通与建模专题）

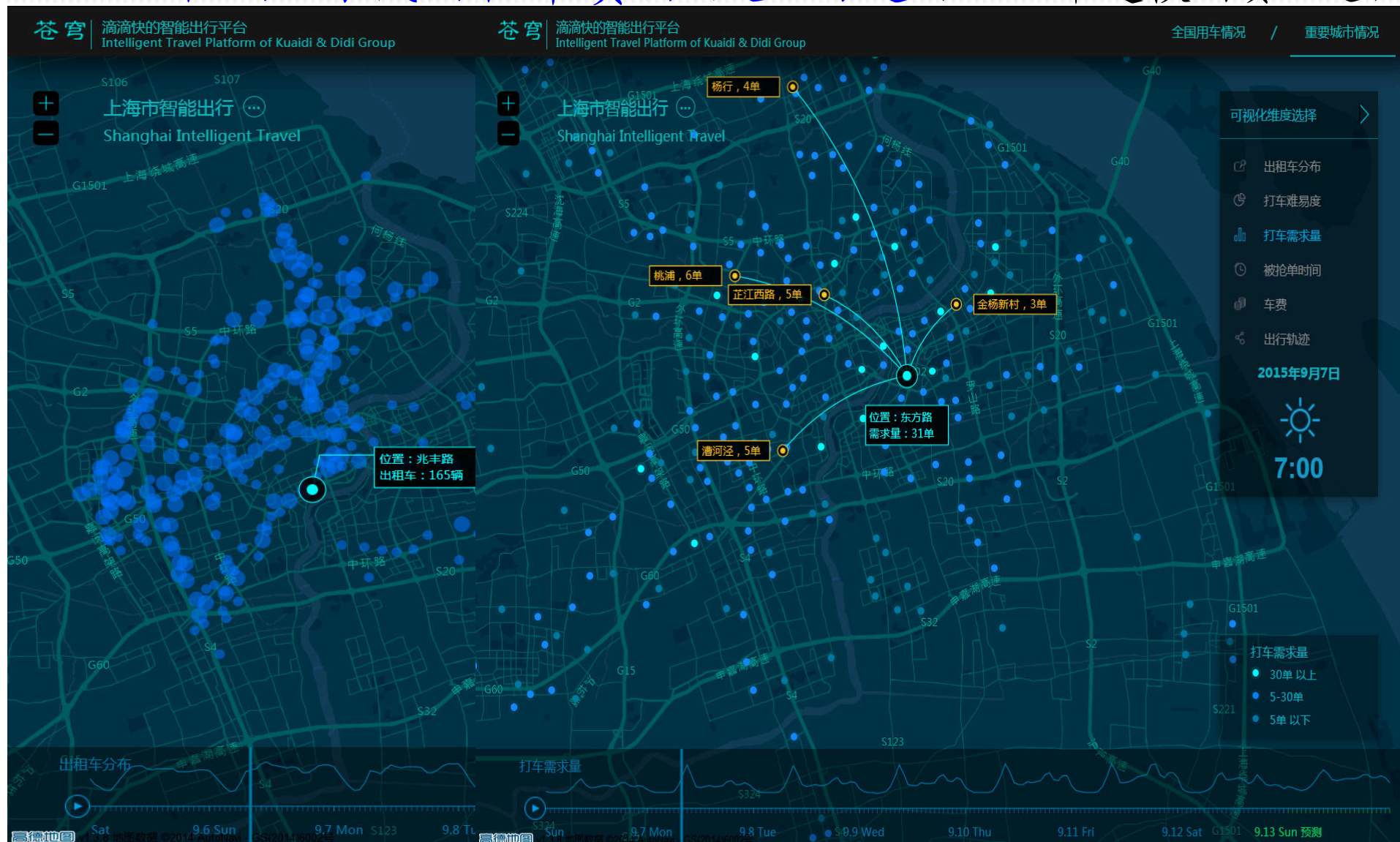
① 数学建模关注交通新问题

数学建模关注城市交通新问题

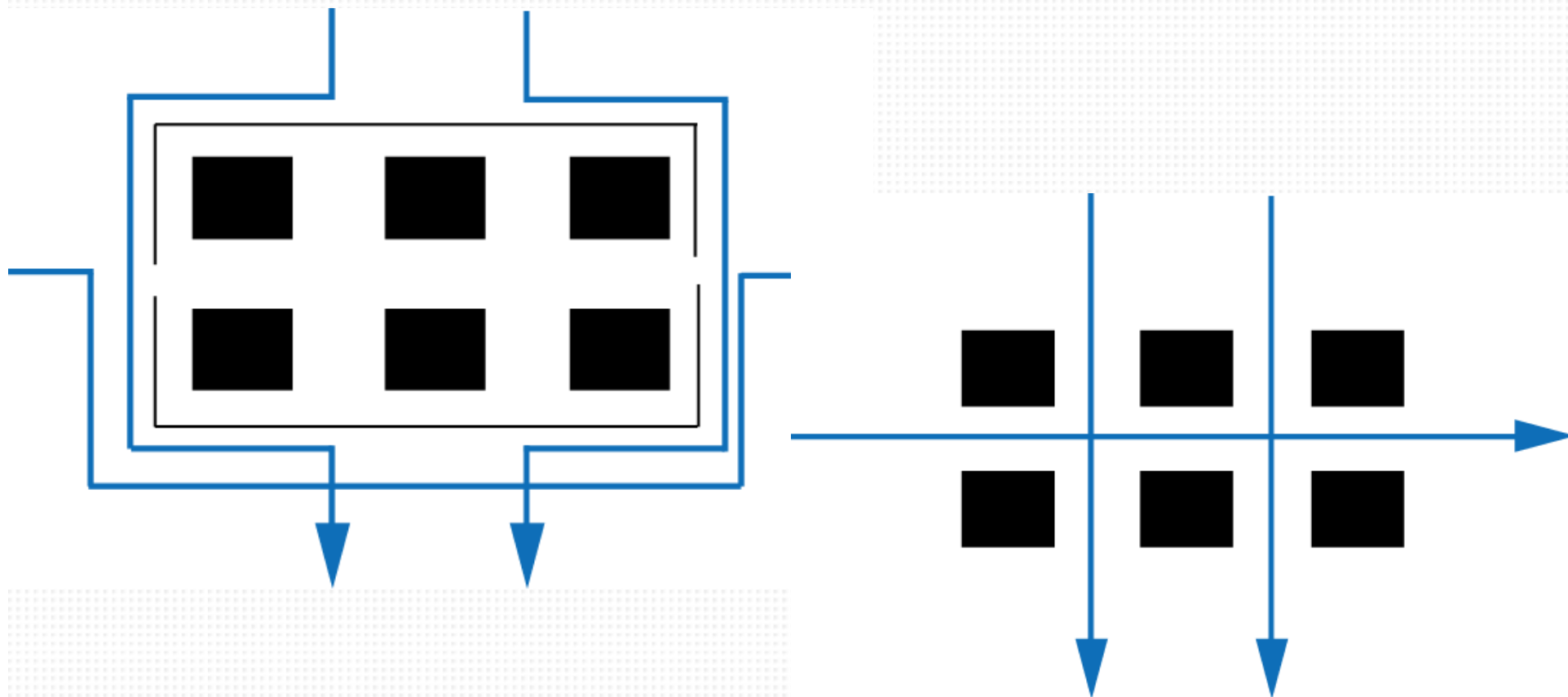


■城市交通就像人身体里的血液循环，**交通拥堵**成为全民关注的焦点

互联网+时代出租车资源配置问题（2015年建模国赛B题）



□小区开放对道路通行的影响（2016年建模国赛B题）



□ 关注城市交通最后一公里：共享单车的可行性



□ 图片来源：左：新华网；右：中国青年报

数学建模关注城市交通新问题

□ 大数据时代下基于多源数据的道路交通流状态重构
(2018年深圳杯数学建模挑战赛题目)

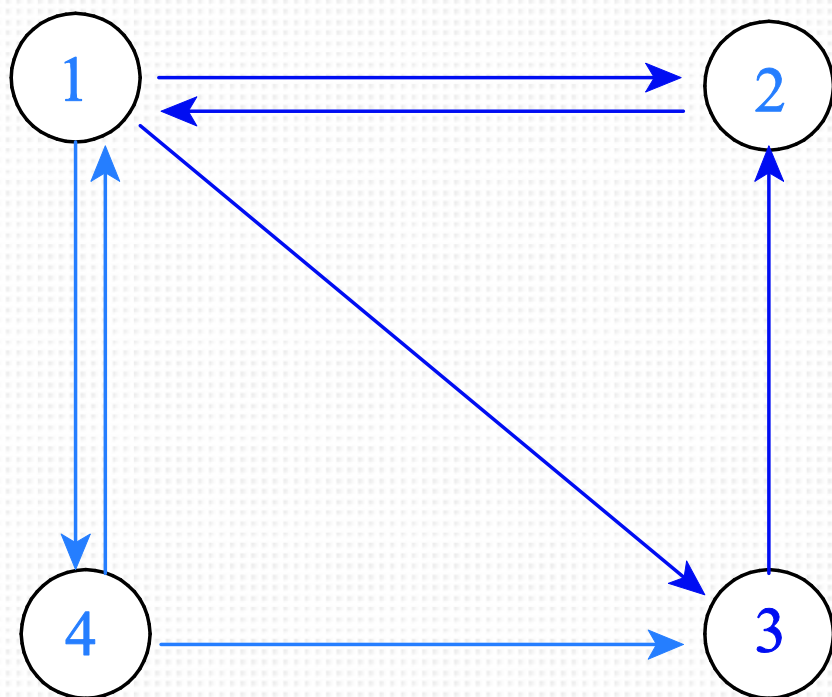
□ 观看奥运会的公交线路选择问题 (2007年建模国赛B题,
设计巧妙, 重点讨论)

□ 数学建模关注交通新问题, 评价新政策的可行性, 研究交通规律和原理, 激发全国大学生数学建模学子的智慧, 为国家政策和社会问题的研究而贡献力量

② 邻接矩阵与航线条数

邻接矩阵与航线条数

□ 设四个城市1、2、3、4间的单向航线如图所示，则可用邻接矩阵表示为



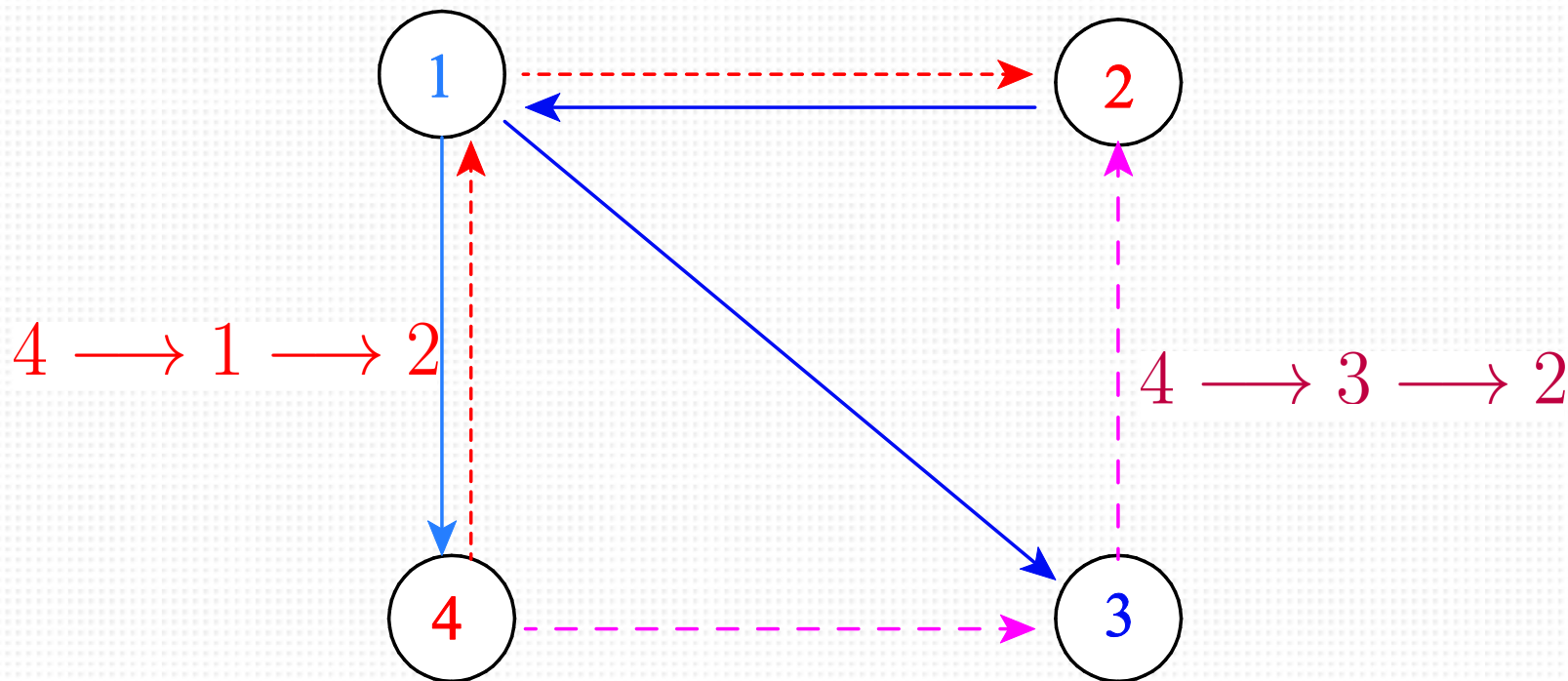
$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

邻接矩阵与航线条数

$$A^2 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \color{red}{1} & \color{red}{0} & \color{red}{1} & \color{red}{0} \end{pmatrix} \begin{pmatrix} 0 & \color{red}{1} & 1 & 1 \\ 1 & \color{red}{0} & 0 & 0 \\ 0 & \color{red}{1} & 0 & 0 \\ 1 & \color{red}{0} & 1 & 0 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & \color{red}{2} & 1 & 1 \end{pmatrix}$$

$$A^2(4, 2) = 1 \times 1 + 0 \times 0 + 1 \times 1 + 0 \times 0 = 2$$

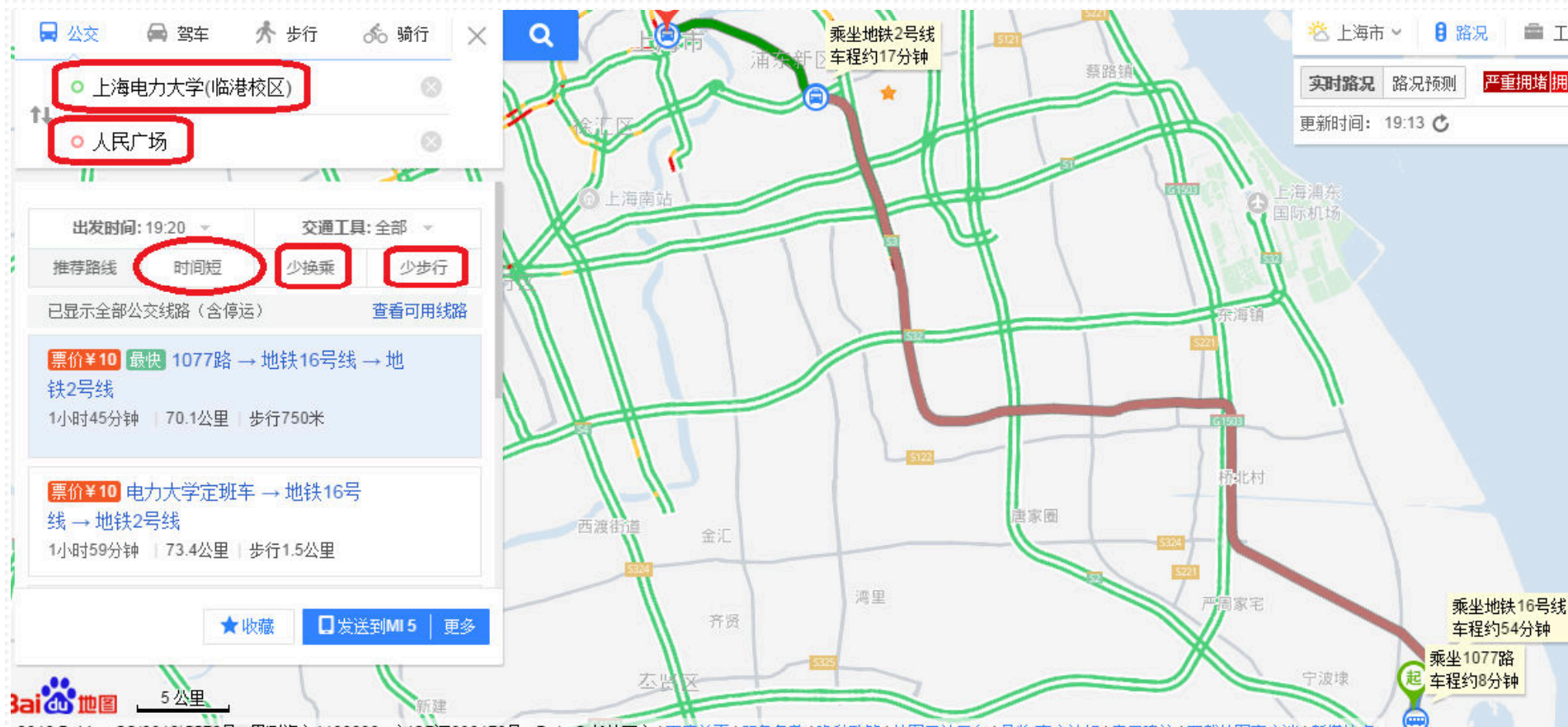
□ 含义：换乘一次



③ 建模案例：公交线路选择问题

公交线路选择问题

□ 出行路线查询：百度地图的日常使用



□ APP给出了最优路线方案，其选择路线的原理是什么？

公交线路选择问题

□引例：图1是由两条公交线路构成的公交网。**实线**表示第1条路线，依次经过站点1、3、4、5，**虚线**表示第2条路线，依次经过站点2、3、5。我们如何研究任意两站点之间的最优路线？

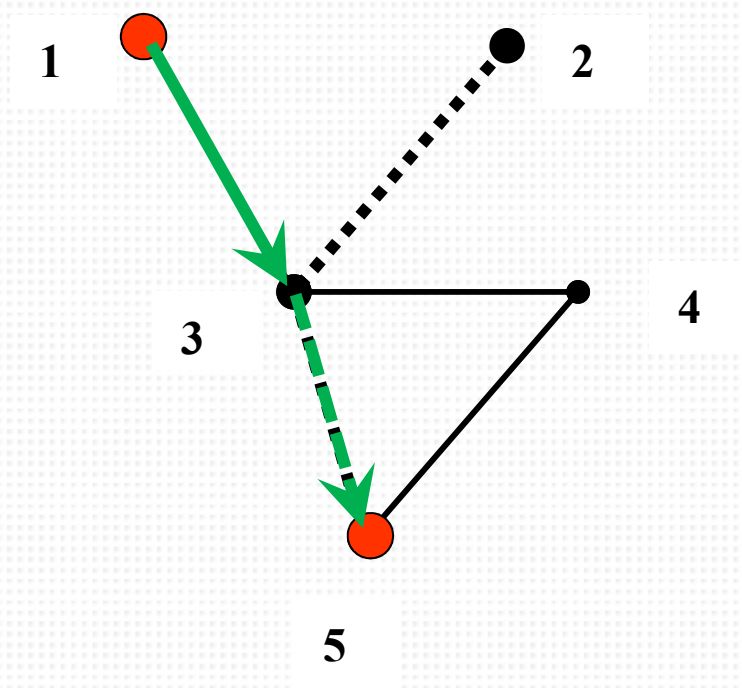
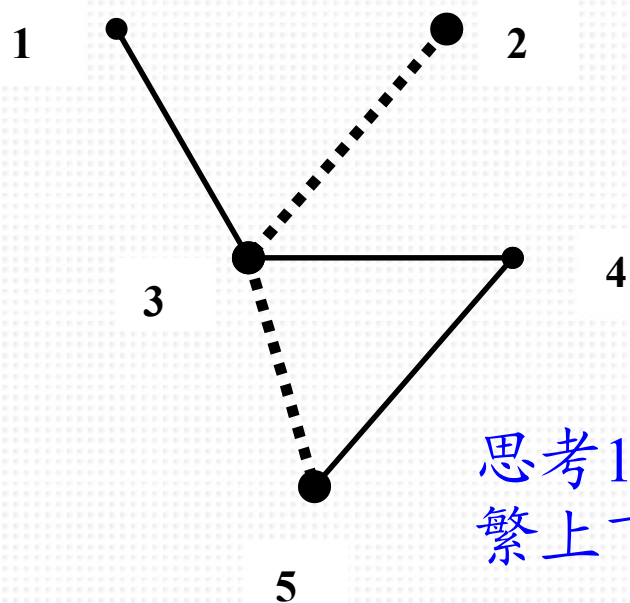


图 1 虚拟公交网

出行时间最短的最优出行路线

□有向赋权图：初始赋权矩阵



$$A = \begin{pmatrix} 0 & \infty & dt & \infty & \infty \\ \infty & 0 & dt & \infty & \infty \\ dt & dt & 0 & dt & dt \\ \infty & \infty & dt & 0 & dt \\ \infty & \infty & dt & dt & 0 \end{pmatrix}$$

思考1：使用传统方法的最优路线可能会出现频繁上下车（换乘几十次）的方案，如何改进？

图1 虚拟公交网

思考2：由于实际公交线有上千条和站点数量达上万个，计算成本太高，如何突破

出行时间最短的最优出行路线

① 建立一步直达出行时间矩阵（每站3分钟, $dt=3$ ）

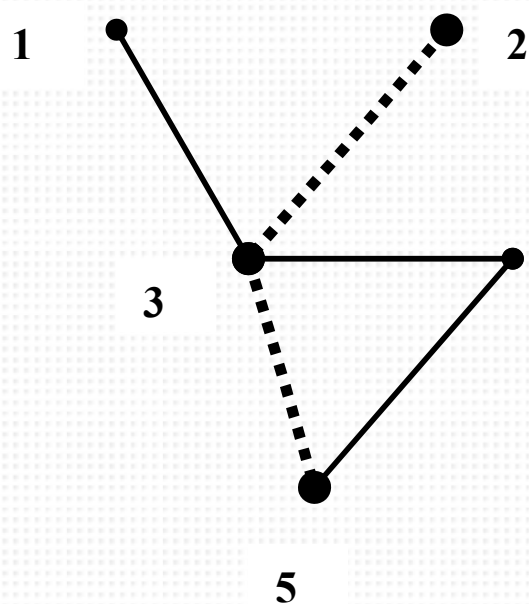


图1 虚拟公交网

$$a_{ij} = \begin{cases} n \cdot dt, & i\text{站与}j\text{站乘某一公交共有}n\text{站的距离} \\ \infty, & i\text{站到}j\text{站不可乘任一公交直接到} \end{cases}$$

$$A = \begin{pmatrix} 0 & \infty & dt & 2dt & 3dt \\ \infty & 0 & dt & \infty & 2dt \\ dt & dt & 0 & dt & dt \\ 2dt & \infty & dt & 0 & dt \\ 3dt & 2dt & dt & dt & 0 \end{pmatrix}$$

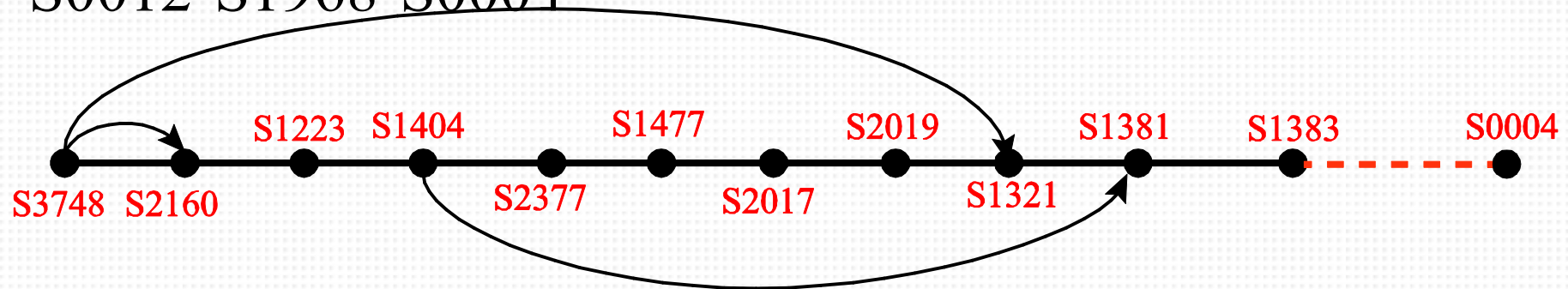
思考3 换乘往往会带来更优出行方案

公交线路选择问题

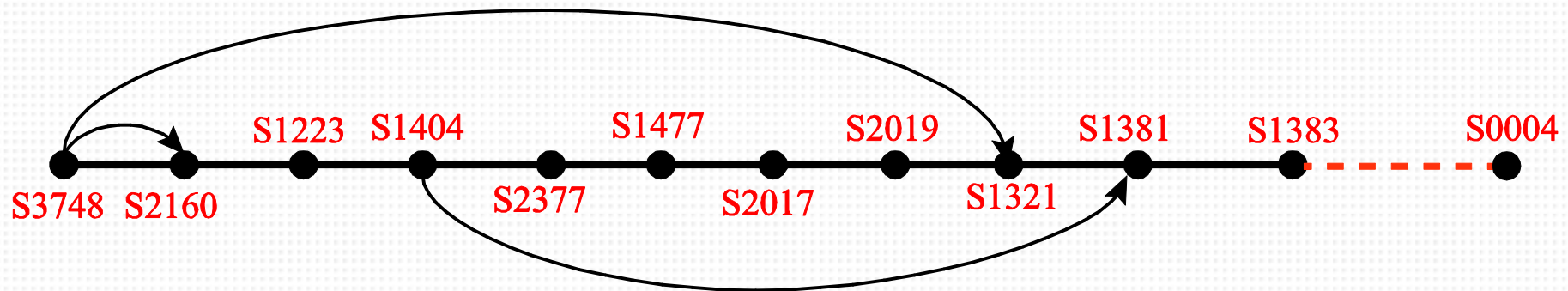
□ 推广到实际路线

Line 002: Stop

S3748-S2160-S1223-S1404-S2377-S1477-S2017-S2019-S1321-S1381-S1383-S0004



公交线路选择问题



$$A(3748, 2160) = 1 \times dt, \quad A(3748, 1321) = 8 \times dt,$$

...

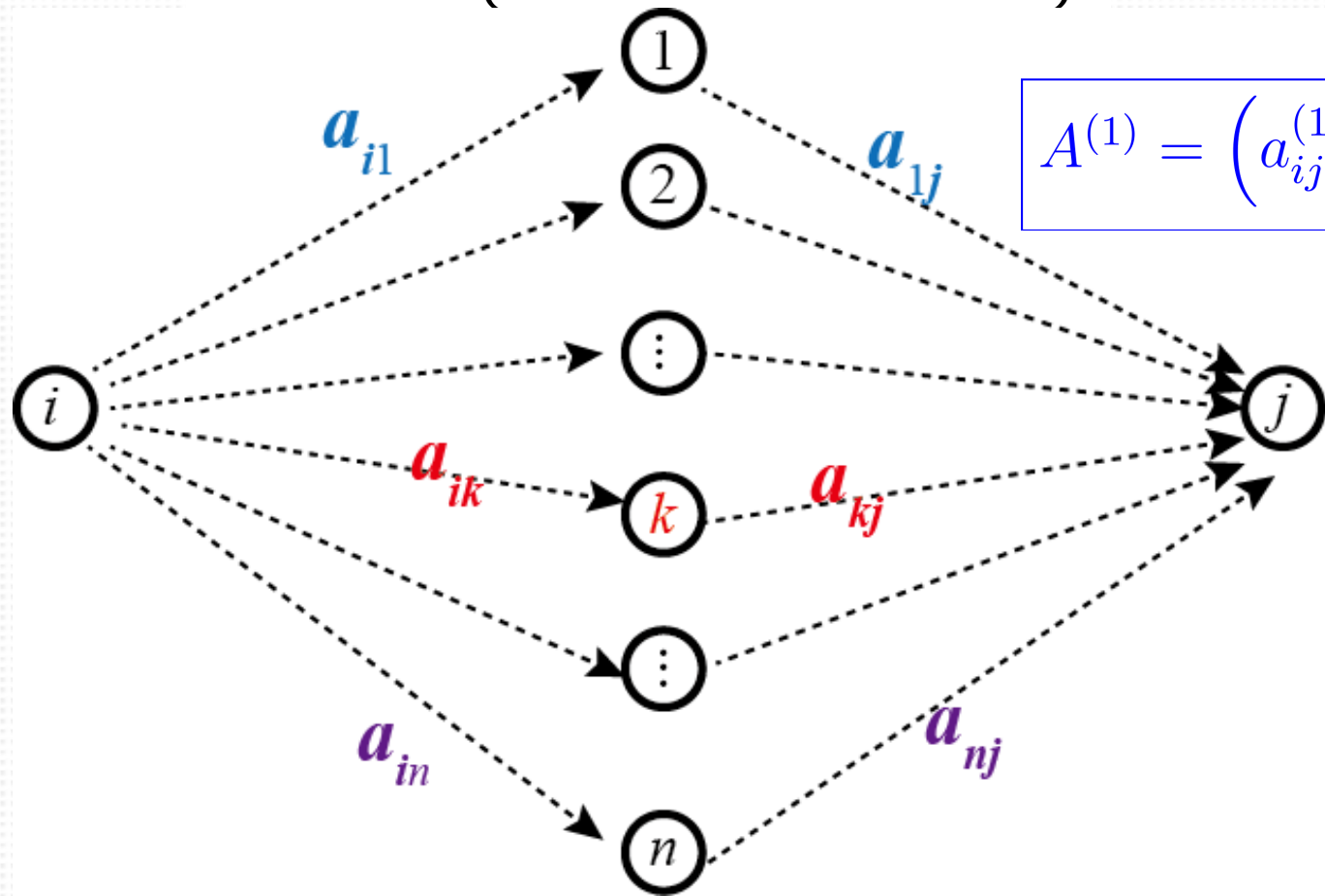
$$A(1404, 1381) = 6 \times dt$$

$$A(\text{Line2}(i), \text{Line2}(j))$$

$$= \min \{ A(\text{Line2}(i), \text{Line2}(j)), (j - i) \times dt \}$$

② 考虑允许**换乘一次**的任意两站点之间的最短时间矩阵:

$$a_{ij}^{(1)} = \min \left\{ \min_{1 \leq k \leq 5} \{a_{ik} + a_{kj}\}, a_{ij} \right\},$$



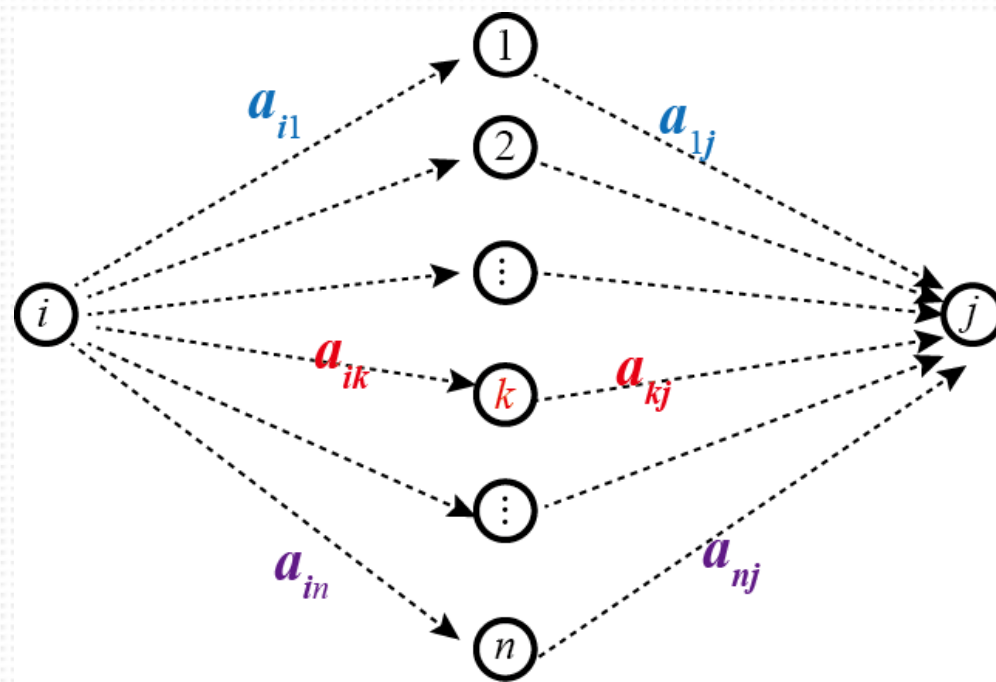
② 考虑允许换乘一次的任意两站点之间的最短时间
矩阵: t_0 为换乘额外等待时间

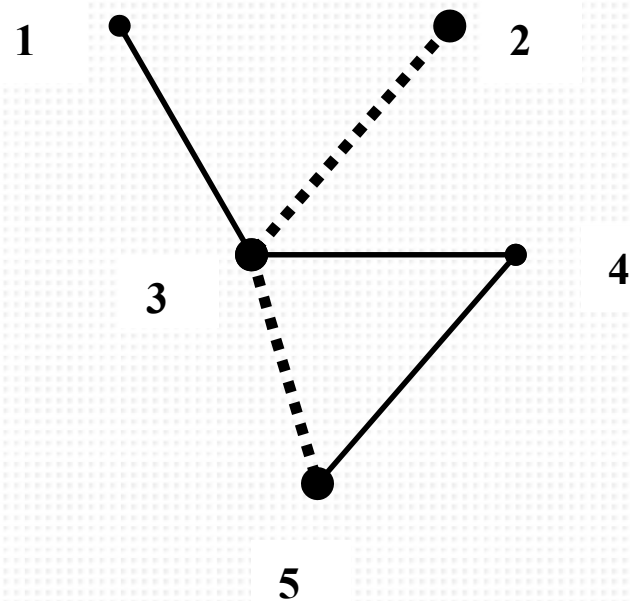
$$a_{ij}^{(1)} = \min \left\{ \min_{1 \leq k \leq n} \{a_{ik} + a_{kj}\} + t_0, a_{ij} \right\},$$

$$\min A(i, :) + A(:, j)^T = \min \{ (a_{i1}, \dots, a_{in}) + (a_{1j}, \dots, a_{nj}) \}$$

$$A^{(1)} = A \otimes A = \left(a_{ij}^{(1)} \right)_{n \times n},$$

称为Floyd矩阵乘法算法





$$A = \begin{pmatrix} 0 & \infty & 3 & 6 & 9 \\ \infty & 0 & 3 & \infty & 6 \\ 3 & 3 & 0 & 3 & 3 \\ 6 & \infty & 3 & 0 & 3 \\ 9 & 6 & 3 & 3 & 0 \end{pmatrix}$$

图1 虚拟公交网

$$A(1, :) + A(:, 5)^T = (0, \infty, 3, 6, 9) + (9, 6, 3, 3, 0) = (9, \infty, 6, 9, 9)$$

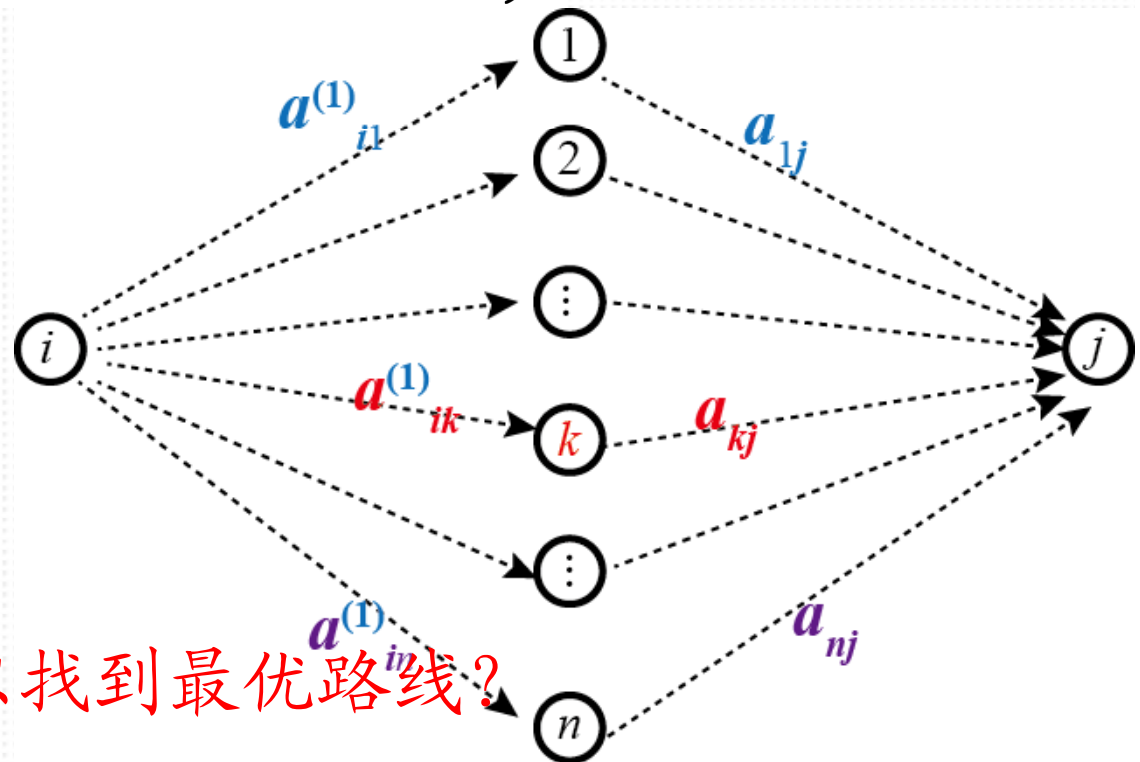
$$t_0 = 2, \quad a_{15}^{(1)} = \min \left\{ \min_{1 \leq k \leq 5} \{a_{1k} + a_{k5}\} + t_0, a_{15} \right\} = 8$$

③ 考虑允许换乘2次的任意两站点之间的最短时间矩阵:

$$A^{(2)} = A \otimes A \otimes A = A^{(1)} \otimes A = \left(a_{ij}^{(2)} \right)_{n \times n}$$

$$a_{ij}^{(2)} = \min \left\{ \min_{1 \leq k \leq n} \left\{ a_{ik}^{(1)} + a_{kj} \right\} + t_0, a_{ij}^{(1)} \right\},$$

思考4: $A^{(2)} * A^{(2)} = A^{(?)}$



思考5: 换乘多少次可以找到最优路线?

乘公交，看奥运

我国人民翘首企盼的第29届奥运会明年8月将在北京举行，届时有大量观众到现场观看奥运比赛，其中大部分人将会乘坐公共交通工具（简称公交，包括公汽、地铁等）出行。这些年来，城市的公交系统有了很大发展，北京市的公交线路已达800条以上，使得公众的出行更加通畅、便利，但同时也面临多条线路的选择问题。针对市场需求，某公司准备研制开发一个[解决公交线路选择问题的自主查询](#)计算机系统。

为了设计这样一个系统，其核心是线路选择的模型与算法，应该从实际情况出发考虑，满足查询者的各种不同需求。请你们解决如下问题：

1、仅考虑公汽线路，给出任意两公汽站点之间线路选择问题的一般数学模型与算法。并根据附录数据，利用你们的模型与算法，求出以下6对起始站→终到站之间的最佳路线（要有清晰的评价说明）。

- | | |
|-----------------|-----------------|
| (1)、S3359→S1828 | (2)、S1557→S0481 |
| (3)、S0971→S0485 | (4)、S0008→S0073 |
| (5)、S0148→S0485 | (6)、S0087→S3676 |

【附录1】基本参数设定

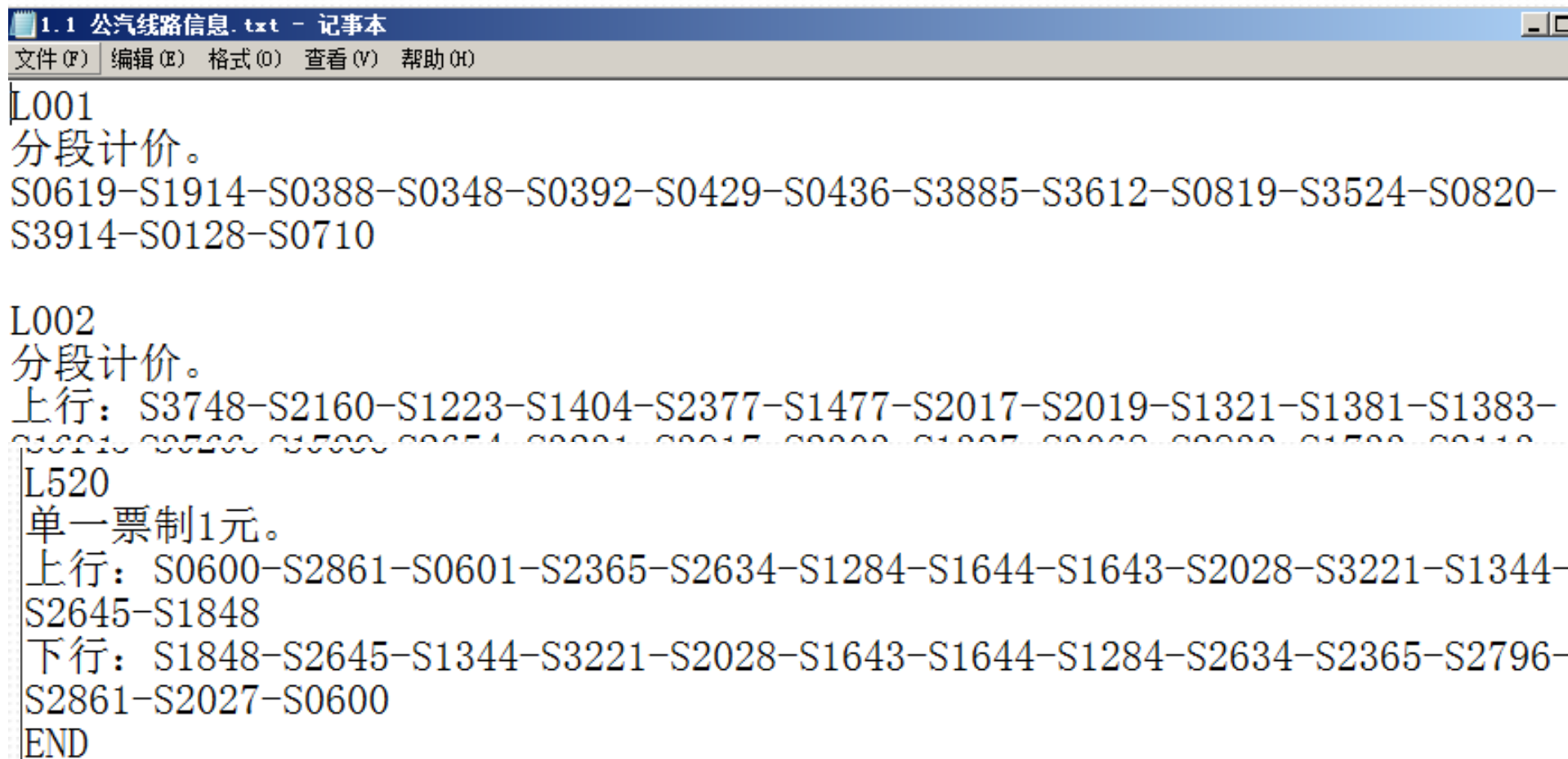
相邻公汽车站平均行驶时间(包括停站时间): 3分钟)

公汽票价: 分为单一票价与分段计价两种, 标记于线路后; 其中分段计价的票价为: 0~20站: 1元; 21~40站: 2元; 40站以上: 3元

注: 以上参数均为简化问题而作的假设, 未必与实际数据完全吻合。

问题1：只考虑公交情形

- ❑ 给出了520条公交线路的信息
- ❑ 初始数据的处理：文字与数字；替换，做标记



1.1 公交线路信息.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

L001
分段计价。
S0619-S1914-S0388-S0348-S0392-S0429-S0436-S3885-S3612-S0819-S3524-S0820-S3914-S0128-S0710

L002
分段计价。
上行：S3748-S2160-S1223-S1404-S2377-S1477-S2017-S2019-S1321-S1381-S1383-S3310-S3520-S1500-S0054-S0001-S0015-S0000-S1005-S0000-S0000-S1500-S0110

L520
单一票制1元。
上行：S0600-S2861-S0601-S2365-S2634-S1284-S1644-S1643-S2028-S3221-S1344-S2645-S1848
下行：S1848-S2645-S1344-S3221-S2028-S1643-S1644-S1284-S2634-S2365-S2796-S2861-S2027-S0600
END

一步直达矩阵的处理

□做标记：没有区分上行和下行：-1；环线：-2

001

0619 1914 0388 0348 0392 0429 0436 3885 3612 0819 3524 0820 3914 0128

0710

-1

002

3748 2160 1223 1404 2377 1477 2017 2019 1321 1381 1383 1691 3766 1729

2654 3231 3917 2303 1327 3068 2833 1733 2113 2636 0012 1968 0004

0004 1968 0012 2636 2113 2112 2833 0618 1327 2303 3917 3231 2654 1729

3766 1691 1383 1381 1321 2019 2017 1477 1404 1223 2160 3748

003

0417 0272 1973 3425 1433 3476 2337 1027 1065 2974 0234 0521 3737 3806

1682 1684 3925 3897 2489 2488

-1

004

3010 0582 0579 1994 1041 0105 1429 2118 1430 3600 2582 1782 0657 3820

一步直达矩阵的处理

□ 没有区分上行和下行，原路返回：抽象成2条线路

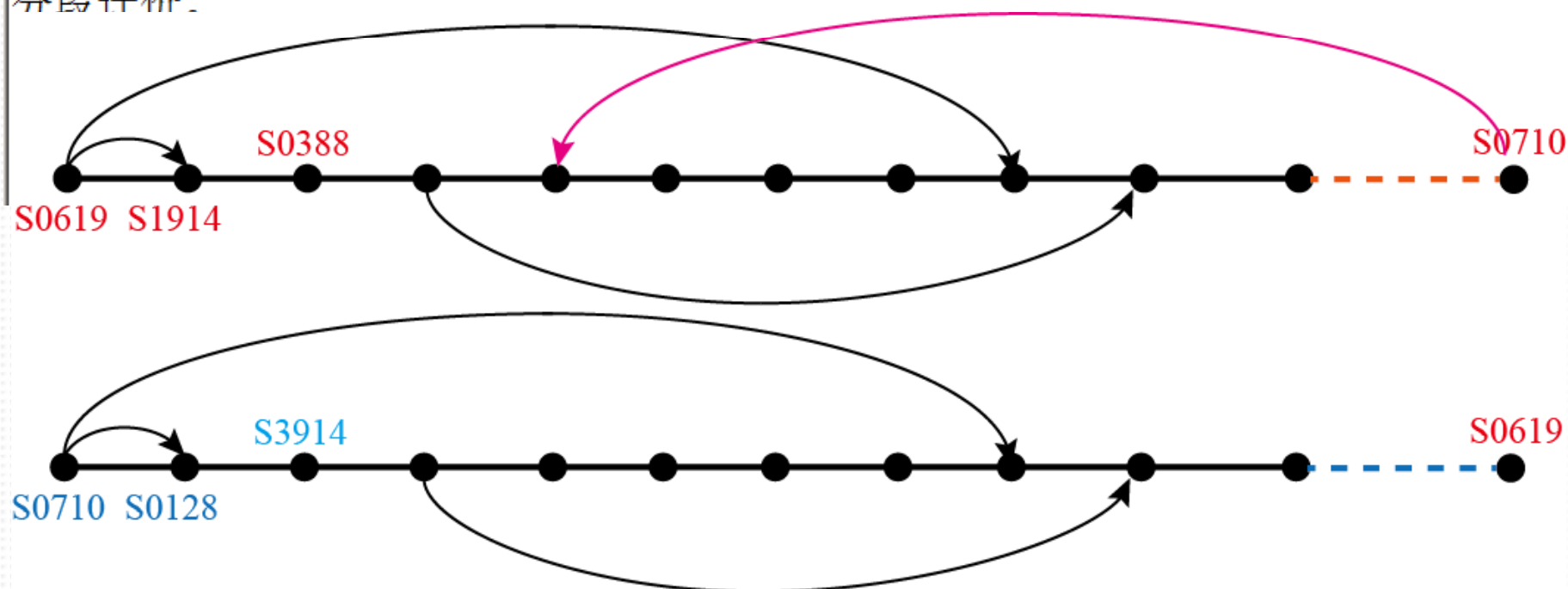
L001

分段计价。

S0619-S1914-S0388-S0348-S0392-S0429-S0436-S3885-S3612-S0819-S3524-S0820-S3914-S0128-S0710

L002

分段计价。



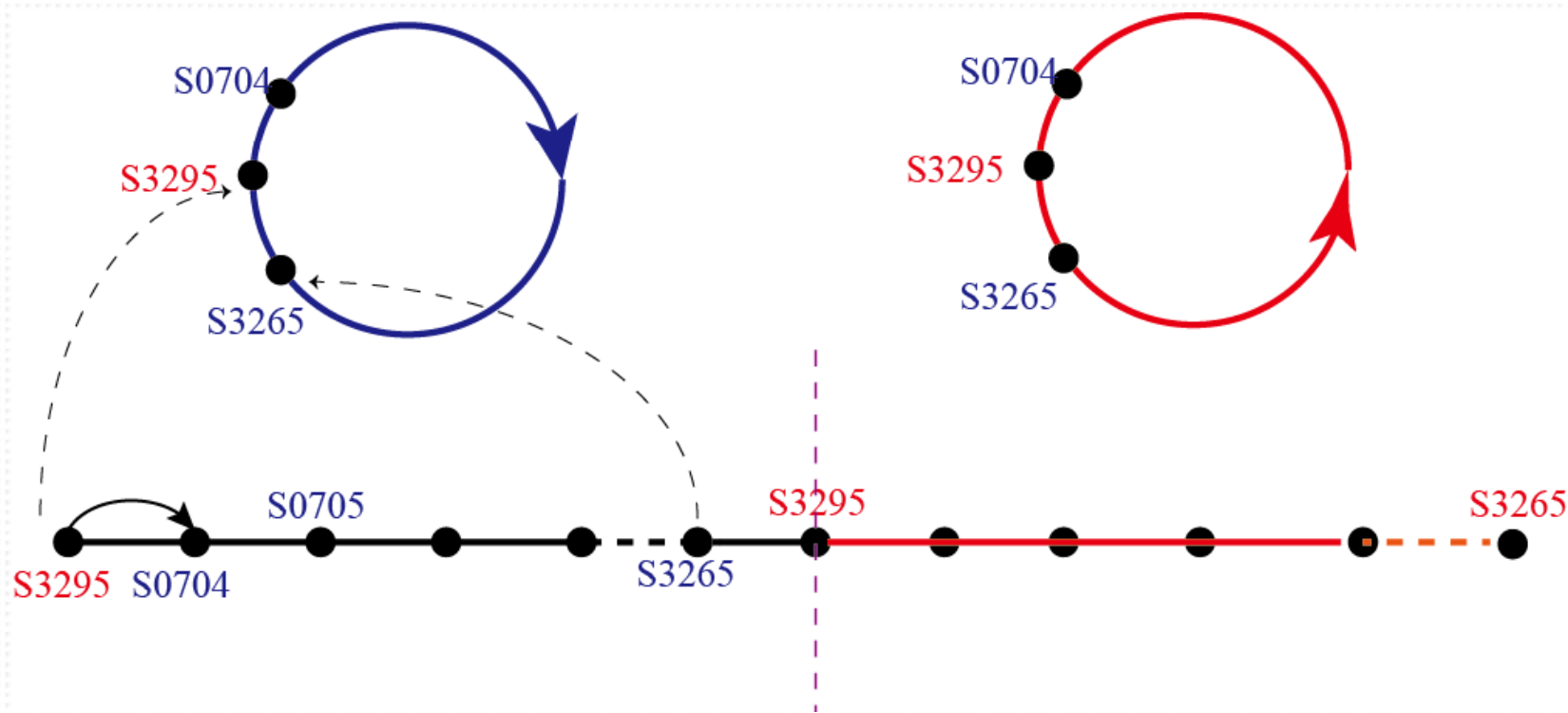
一步直达矩阵的处理

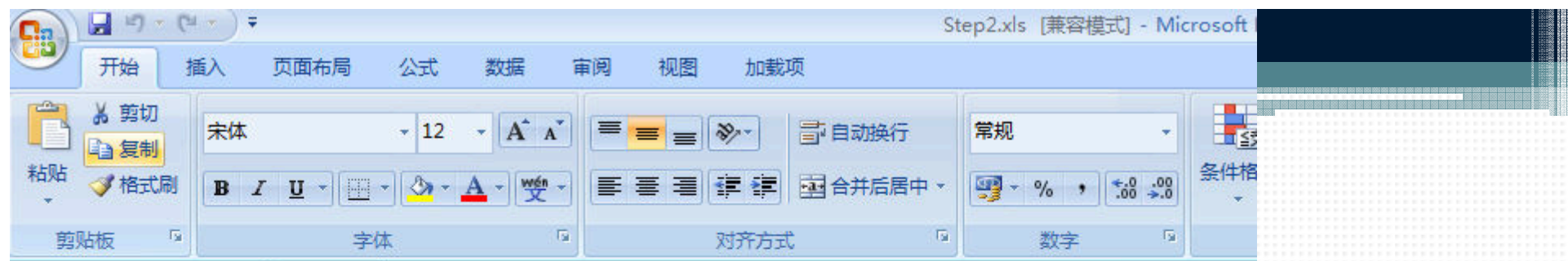
□ 环线路线:

L089

单一票制1元。

环行: S3295-S0704-S0705-S1337-S1831-S1749-S0314-S1734-S0532-S0325-S0317-S0327-S1341-S2285-S1333-S1334-S3265-S3295





I16 3042

1	1	0619	1914	0388	0348	0392	0429	0436	3885	3612	0819	3524	0820	3914	0128
2		0710													
3	619	-1													
4	-1														
5	2	3748	2160	1223	1404	2377	1477	2017	2019	1321	1381	1383	1691	3766	1729
6		2654	3231	3917	2303	1327	3068	2833	1733	2113	2636	0012	1968	0004	
7	3748	0004	1968	0012	2636	2113	2112	2833	0618	1327	2303	3917	3231	2654	1729
8	4	3766													
9	3	0417													
10		1682													
11	417	-1													
12	-1														
13	4	3010													
14		3042													
15	3010														

Variable Editor - Line										
File Edit View Graphics Debug Desktop Window Help										
Stack: Base No valid plots for: Li...										
Line <1040x144 double>										
	1	2	3	4	5	6	7	8	9	10
1	619	1914	388	348	392	429	436	3885	3612	819
2	710	128	3914	820	3524	819	3612	3885	436	429
3	3748	2160	1223	1404	2377	1477	2017	2019	1321	1381
4	4	1968	12	2636	2113	2112	2833	618	1327	2303
5	417	272	1973	3425	1433	3476	2337	1027	1065	2974
6	2488	2489	3897	3925	1684	1682	3806	3737	521	234
7	3010	582	579	1994	1041	105	1429	2118	1430	3600
8	679	2184	2182	992	2324	2323	1957	658	3042	3820
9	2817	1961	3171	925	2119	1789	1791	457	1771	1793
10	1236	2086	2038	3264	477	130	684	405	3453	751
11	43	977	1110	3403	528	605	2846	409	3543	3760
12	2915	3086	2913	1052	2018	2015	1122	1123	3029	3356
13	2135	574	1016	1018	2058	3274	3593	2859	2473	3486
14	2864	1846	3677	1972	1923	2661	235	2394	2908	294
15	1037	2185	1248	666	665	1354	2193	1353	1691	2585

一步直达矩阵的处理

□ 扫描每一条路线，得到一步直达矩阵

$$a_{ij}^0 = \begin{cases} 0, & i = j; \\ \infty, & i, j \text{ 不在 } B \text{ 的同一行}; \\ \min_k 3 |k_l - k_m|, & i, j \text{ 同时在 } B \text{ 的第 } k \text{ 行的第 } k_l, k_m \text{ 列}. \end{cases}$$

Variable Editor - A1

File Edit View Graphics Debug Desktop Window Help

Stack: Base No valid plots for: A1...

A1 <3957x3957 double>

	1	2	3	4	5	6	7
1	0	800	800	800	800	800	800
2	800	0	800	72	800	800	6
3	800	800	0	800	800	800	800
4	800	72	800	0	144	141	66
5	800	800	800	150	0	3	800
6	800	800	800	147	3	0	800
7	800	6	800	66	800	800	0
8	800	800	800	800	800	800	800
9	800	800	800	800	800	800	800
10	800	800	800	800	800	800	800
11	800	800	800	800	800	800	800
12	800	800	800	6	800	800	800
13	800	800	800	800	800	800	800
14	800	800	800	800	800	800	800
15	800	800	800	800	800	800	800

换乘一次的最短时间矩阵

□编写改进的Floyd矩阵算法

```
- function bus_route_main()
-   load result_A0 A0
-   [A1, pathA1]=Bus_route(A0, A0); %换乘一次最优时间矩阵
-   [A2, pathA2]=Bus_route(A1, A0); %换乘2次最优时间矩阵

- function [A2, pathA2]=Bus_route(A1, A0)
-   m=length(A0); %公交站点的数量
-   cost=5; %换乘公汽耗时5分钟
-   A2=zeros(m, m); %换乘后最短时间矩阵初始化
-   pathA2=zeros(m, m); %存储下换乘点矩阵
-   for i=1:m
-       for j=1:m
-           temp=A1(i, :)+A0(:, j)';
-           [temp, k]=min(temp);
-           A2(i, j)=min(temp+cost, A1(i, j));
-           if A2(i, j)<800 && A2(i, j)<A1(i, j)
-               pathA2(i, j)=k;
-           end
-       end
-   end
-   save result_A2 A2 pathA2
```

问题2：同时考虑公交和地铁情形

□2条地铁线路

T1

票价3元，本线路使用，并可换乘T2。

D01-D02-D03-D04-D05-D06-D07-D08-D09-D10-D11-
D12-D13-D14-D15-D16-D17-**D18**-D19-D20-D21-D22-
D23

T2

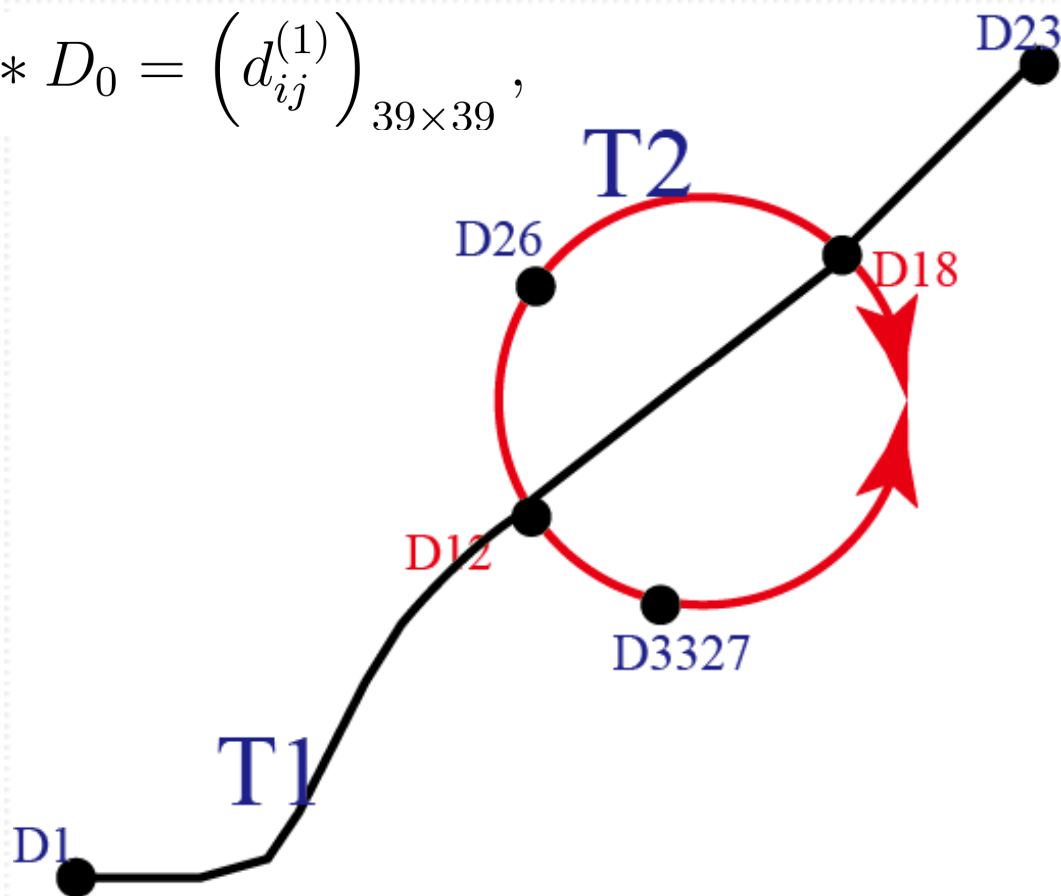
票价3元，本线路使用，并可换乘T1。

环行：D24-D25-D26-**D12**-D27-D28-D29-D30-D31-
D32-**D18**-D33-D34-D35-D36-D37-D38-D39-D24

同时考虑公交和地铁情形

□ 2条地铁间任意地铁站点D之间的最短时间矩阵

$$D^{(1)} = D_0 * D_0 = \left(d_{ij}^{(1)} \right)_{39 \times 39},$$



同时考虑公交和地铁情形

□地铁与公交的关系

2.1 地铁T1线换乘公交信息.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

D01: S0567, S0042, S0025
D02: S1487
D03: S0303, S0302
D04: S0566
D05: S0436, S0438, S0437, S0435
D06: S0392, S0394, S0393, S0391
D07: S0386, S0388, S0387, S0385
D08: S3068, S0617, S0619, S0618, S0616
D09: S1279
D10: S2057, S0721, S0722, S0720
D11: S0070, S2361, S3721
D12: S0609, S0608
D13: S2633, S0399, S0401, S0400
D14: S3321, S2535, S2464
D15: S3320, S2534

2.2 地铁T2线换乘公交信息.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

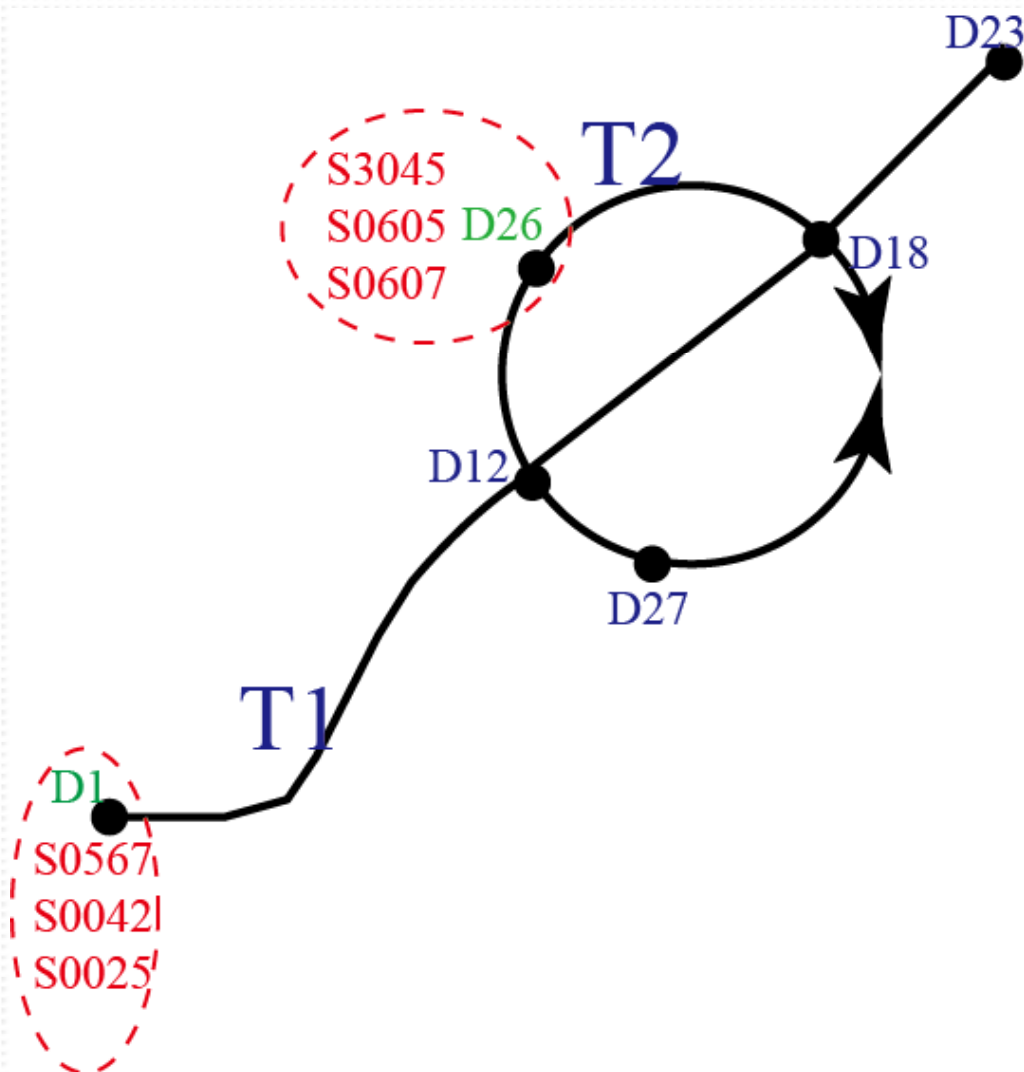
D24: S0537, S3580
D25: S0526, S0528, S0527, S0525
D26: S3045, S0605, S0607
D12: S0609, S0608
D27: S0087, S0088, S0086
D28: S0855, S0856, S0854, S0857

同时考虑公交和地铁情形

□ 用地铁站之间的距离替代地铁区域内公共汽车的距离，然后比较任两站之间的距离与原初始赋权可达矩阵 A^0 ，并取较小值就可以形成新的包含地铁的初始赋权可达矩阵

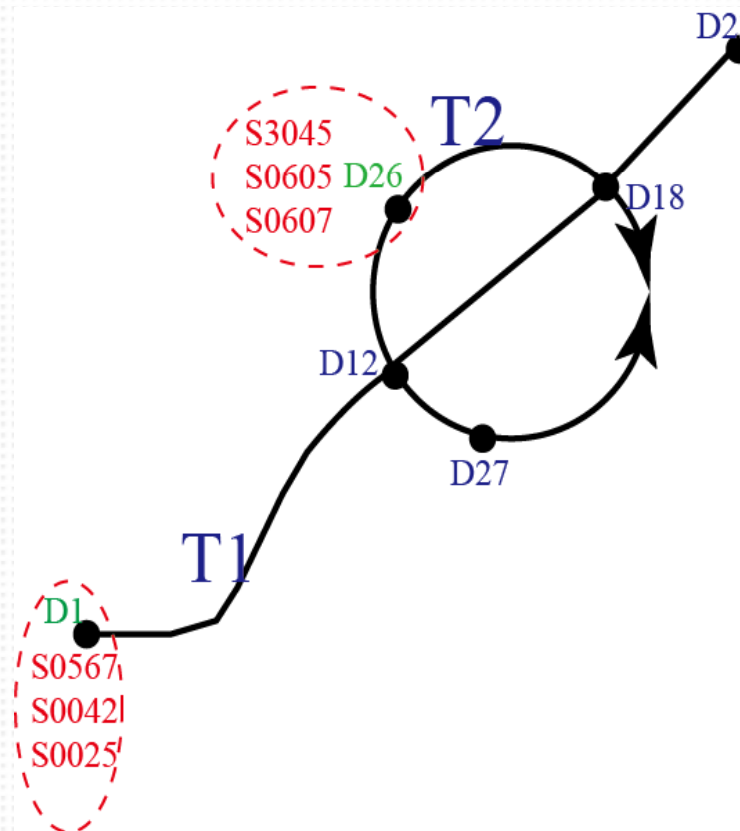
$T^{(0)}$

$$t_{567,3045}^{(0)} = \min\{a_{567,3045}^{(0)}, d_{1,26}^{(1)}\}$$



同时考虑公交和地铁情形

- 换乘一次、两次.....
- 不必换乘很多次：任意两站经**五次**换乘后，已经98%可达；
- 换乘一次的计算量：至少50分钟
- 不同换乘时间的影响：微调



同时考虑公交和地铁情形

表 3 问题 (2) 最佳路线结论

起始至终点	总时间(分钟)	换乘线(路长)	中转站	换乘线(路长)	中转站	换乘线(路长)	中转站	换乘线(路长)	中转站	换乘线(路长)
S3359	64	L0324(3) L0484(3)	S1746	L0485(14)	S1784	L0167(1) L0217(1)	/	/	/	/
S1828	59	L0324(3) L0469(3) L0484(3)	S2027	L0201(3)	S0609	D12 → D37	S1961	L0428(2)	S1671	L0041(1)
S1557	106	L0084(12) L0363(12)	S1919	L0189(3)	S3186	L0460(17)	/	/	/	/
S0481	99	L0084(12) L0363(12)	S1919	L0189(3)	S3186	L0091(10) L0317(10)	S00902	L0254(3) L0312(3) L0447(3) L0460(3) L0516(3)	/	/
S0971	93	L0094(6) L0119(6)	S0567	D1 → D21	S0464	L0104(5) L0395(5)	/	/	/	/
S0485	92	L0094(6) L0119(6)	S0567	D1 → D15	S2534	L0156(5)	S2210	L0417(3)	/	/
S0008	53.5	L0200(6)	S2534	D15 → D25	S0525	L0103(2)	/	/	/	/

问题3：同时考虑公交、地铁和步行

- 步行也是一种交通工具
- 提供了新的路线选择
- 任意两个站点还有一个成本：步行时间
- 步行优就选步行
- 推广：共享单车

问题3：同时考虑公交、地铁和步行

设任意站点之间的步行时间矩阵记为 W^* ，同时考虑公交和地铁线路情形中的最优赋权矩阵记为 A^* ，比较 A^* 和 W^* 的对应元素，取较小的值，可得考虑步行时的初始赋权可达矩阵 $F^{(0)} = (f_{ij}^{(0)})$ ，其中

$$f_{ij}^{(0)} = \begin{cases} w_{ij}^*, & a_{ij}^* > w_{ij}^* \\ a_{ij}^*, & w_{ij}^* \leq a_{ij}^* \end{cases}$$

再对 $F^{(0)}$ 作 Floyd 矩阵乘法运算，分别得到至多经过 k 次换乘的最优路径矩阵

$F^{(k)} = F^{(k-1)} \otimes F^{(0)}, k = 0, 1, \dots$ 。不断增加换乘次数，直到 $F^{(k)}$ 和 $F^{(k+1)}$ 相等，取 $F^{(k)}$

为全局最优矩阵 F^* 。

进一步讨论

- 出行费用最省的最优路线
- 把计算出的结果存储到数据库，供用户查询使用
- 拥堵对出行路线的影响：结合实时拥堵信息，对路线调整和动态更新
- 大数据预测：对不同时刻出发情形进一步更新和预测
- 小结：该问题设计巧妙，解决思路递进，有助于学生动手实践和创新思维训练，推荐认真研究并动手实现



Thanks !