

第三届数学中国数学建模网络挑战赛

地址：内蒙古数学会
电话：0471-4343756

邮编：010021

网址：www.tzmcm.cn
Email：2010@tzmcm.cn

第三届“ScienceWord 杯”数学中国

数学建模网络挑战赛

承 诺 书

我们仔细阅读了第三届“ScienceWord 杯”数学中国数学建模网络挑战赛的竞赛规则。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的，如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们将受到严肃处理。

我们允许数学中国网站(www.madio.net)公布论文，以供网友之间学习交流，数学中国网站以非商业目的的论文交流不需要提前取得我们的同意。

我们的参赛报名号为：

参赛队员（签名）：

队员 1：

队员 2：

队员 3：

参赛队教练员（签名）：

参赛队伍组别：

第三届数学中国数学建模网络挑战赛

地址：内蒙古数学会
电话：0471-4343756

邮编：010021

网址：www.tzmcm.cn
Email：2010@tzmcm.cn

第三届“ScienceWord 杯”数学中国

数学建模网络挑战赛

编号专用页

参赛队伍的参赛号码：（请各个参赛队提前填写好）：

竞赛统一编号（由竞赛组委会送至评委团前编号）：

竞赛评阅编号（由竞赛评委团评阅前进行编号）：

第三届数学中国数学建模网络挑战赛

地址：内蒙古数学会
电话：0471-4343756

邮编：010021

网址：www.tzmcm.cn
Email: 2010@tzmcm.cn

2010 年第三届“ScienceWord 杯”数学中国 数学建模网络挑战赛

题 目：交通网络中 Braess 悖论的实证分析及其多主体模拟

关 键 词：Braess 悖论；交通规划；非合作均衡；多主体模拟

摘 要：

Braess 悖论指出局部的改善反而可能降低交通网络的整体效率。本文构建了一个非合作均衡模型用于检测交通网络中的 Braess 悖论，并且结合实证数据在北京市展览馆路处发现了 Braess 悖论。进一步，我们采用多主体建模的方法讨论了引入 GPS 导航系统对整个交通网络的影响。结果表明 GPS 导航系统确实能够显著缓解已有的交通阻塞现象，但是要求现有的交通网络承载足够的车流量，否则会适得其反。另外，我们发现 GPS 导航系统在提高整个系统效率的同时能够区别现有网络中的高效线路和低效线路，进而为交通系统的科学管理提供有力支持。

参赛队号 1214

参赛密码
(由组委会填写)

所选题目 B 题

第三届数学中国数学建模网络挑战赛

地址：内蒙古数学会
电话：0471-4343756

邮编：010021

网址：www.tzmcm.cn
Email：2010@tzmcm.cn

英文摘要（选填）

Braess paradox demonstrated a situation in which a new link, added to improve the traffic condition, results in a worse one. In this paper, we proposed a non-cooperative model to detect the paradox. Empirical investigations suggest that such paradox does happen around Zhan Lan Guan Road in Beijing. Moreover, a Multi-Agent based model is built to investigate whether GPS can improve the traffic congestion. The simulation results indicate that GPS will obtain significant progress when it satisfies a certain condition determined by the current traffic network itself. Meanwhile, it is possible for GPS in our model to find out the roads widely used. This can provide scientific support to the traffic management system.

1. 问题的重述

随着中国城市化的发展，城市交通网络的压力越来越大，出行的人们会发现城市的机动车数目有了明显的增加，北京市机动车数量近 400 万[1]。那么如何使得城市的交通网络更加有效率呢？地理学家曾经花费了很多的笔墨讨论区域连通性或者城市的通达性问题。根据地理工作者的直观理解，道路越多，连通性肯定也就越好，通达性势必越强[2]。这样是基于线性假设，根据线性系统的特征，道路越多一定会增加交通网的连通性，但是城市和区域是复杂的非线性系统[3]，而对于非线性系统，增加道路可能会导致道路连通性的下降，从而降低整个系统的效率。

1968 年，意大利数学家 Dietrich Braess 在他的一篇论文中提出了在交通网络中存在悖论[4]。通常情况下，人们在直观上认为在原本拥挤的道路上加入一条新的道路可以使得运输系统的效率增加，但是 Braess 证明了在某些情况下，在交通网络中加入一条通路反而会使得所有用户的出行时间变长从而降低系统的效率。这一违背常理的现象称为 Braess 悖论。这一想法对现实产生了深远的意义，它不仅被应用于交通运输，在其他科学领域里它也成了科学家们必须要讨论的重要部分[5, 6, 7]。由于这一悖论的存在，也使得我们重新考虑是否应该在原有交通网络中加入新道路。

在一个由四条路组成的简易交通网络中，如图 1。在这里，只有一个起点 O 和一个终点 R ，这个图与 Braess 和很多其他学者研究的图形是一致的。左边的图表示最初的交通网络，在这里，边 PQ 是没有加入到网络中的，右边的图表示在原有网络中加入新道路 PQ 。在原始的交通网络中，从起点到终点有两条路线 OPR 与 OQR ；加入新通路 PQ 后，从起点到终点就存在三条路线 OPR 、 OQR 与 $OPQR$ 。Braess 研究了 this 交通网络后发现，加入新通路 PQ 后，整个网络的效率不可思议的下降了，这与我们的直觉违背。

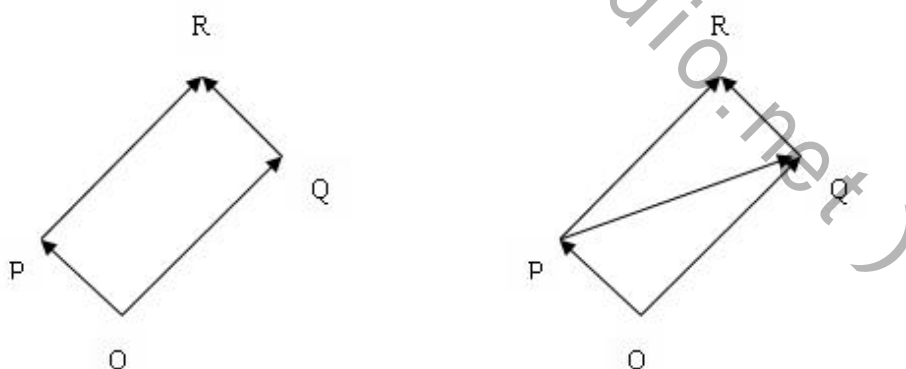


图 1: Braess 悖论示意图

根据博弈论的观点，每一个局中人都关注自己个人的利益而不是系统整体的利益，因此每一个人在追求自己出行时间最短的同时，有可能延长了很多其他人的出行时间，从而使得整个交通网络的效率下降，产生 Braess 悖论。

根据以上的说明，本文主要考虑两个问题：1，在北京二环路以内（包括二环路），交通拥堵的情况是否由 Braess 悖论引起的，即是否存在实际数据表明由

Braess 悖论引起了这一路段的交通拥堵。2, 如果司机广泛使用能够反映当前路况信息的 GPS 系统, 是否能缓解当前的交通压力。

2. 基本假设和符号说明

- i. 假设通过道路 (i, j) 所需时间为 $t_{ij} = a_{ij} + b_{ij}f_{ij}$, 其中 t_{ij} 为道路 (i, j) 上的通行时间, a_{ij} 为道路 (i, j) 上的自由交通时间, 反应了道路的长短; b_{ij} 为道路 (i, j) 的延迟系数(在道路每增加一个单位车流量所增加的通行时间), 反应了道路的宽窄; f_{ij} 表示单位时间内通过道路 (i, j) 的车流量。
- ii. f_k 和 t_k 分别表示经过路径 k 的车流量和通过路径 k 所需的时间。
- iii. T_1 和 T_2 分别表示新增道路前后所有车辆通过系统的总通行时间。
- iv. 每一个体总是希望自己能够尽快通过网络, 即每一个个体总是考虑自己的利益而非总体的利益。
- v. 安装 GPS 可以使得司机观察到道路的当前车流量, 而没有安装 GPS 不能观察到道路的当前车流量。

3. 问题一模型的建立与求解

(一) 模型的建立

对于问题一我们关心在北京二环路以内(包括二环路)的交通拥堵是否由于 Braess 悖论所引起, 即是否存在一条路使得加入这条路之后交通状况变的更加糟糕。对于图 1 所表示的交通网络在实际中并不常见, 针对这个问题, 我们考虑如图 2 所示的交通网络, 这是在北京交通网络中经常出现的结构, 可以很好的代表北京市的交通网络。在这个问题中, 我们只对这样的小单元进行分析, 这样的分析可以推广至北京市二环内部的所有道路。

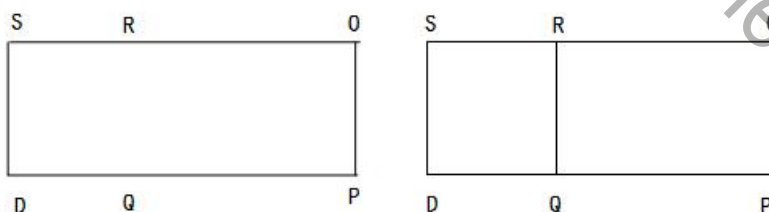


图 2: 实际交通系统中常见的交通网络模型图, 左图表示没有加入新道路时的状态, 右图表示加入了新道路后的交通网络。

假设有 Q 辆车从起点 O 出发, 终点是 D 。对于图 2 中的左图, 即在原始交通网络中, 没有加入新道路 RQ , 车辆从 O 点出发, 到达终点 D 共有两条路线: OPD

和 OSD ，经过路径 OSD 需要的时间如下： $t_{OSD} = t_{OR} + t_{RS} + t_{SD}$ ，其中

$$\begin{cases} t_{OR} = a_{OR} + b_{OR}f_{OR} \\ t_{RS} = a_{RS} + b_{RS}f_{RS} \\ t_{SD} = a_{SD} + b_{SD}f_{SD} \end{cases};$$

经过路径 OPD 的需要的的时间如下： $t_{OPD} = t_{OP} + t_{PQ} + t_{QD}$ ，其中

$$\begin{cases} t_{OP} = a_{OP} + b_{OP}f_{OP} \\ t_{PQ} = a_{PQ} + b_{PQ}f_{PQ} \\ t_{QD} = a_{QD} + b_{QD}f_{QD} \end{cases}。$$

定义路径 OSD 为路径 1， OPD 为路径 2，则

$$\begin{cases} t_1 = t_{OR} + t_{RS} + t_{SD} \\ t_2 = t_{OP} + t_{PQ} + t_{QD} \\ Q = f_1 + f_2 \end{cases}$$

对于图 2 中右图所示的情况，为了缓解交通拥堵，在 RQ 点之间加入新的道路 (R, Q) ，此时从起点 O 到终点 D 有三条路线 $OPQD$ ， $ORSD$ 和 $ORQD$ 。经过路径 $ORQD$ 的时间如下： $t_{ORQD} = t_{OR} + t_{RQ} + t_{QD}$ ，其中

$$\begin{cases} t_{OR} = a_{OR} + b_{OR}f_{OR} \\ t_{RQ} = a_{RQ} + b_{RQ}f_{RQ} \\ t_{QD} = a_{QD} + b_{QD}f_{QD} \end{cases};$$

其余两条路径的时间表达式与上面的路径 1 和路径 2 一致。定义 $ORSD$ 为路径 1， $ORQD$ 为路径 2， $OPQD$ 为路径 3，则

$$\begin{cases} t_1 = t_{OR} + t_{RS} + t_{SD} \\ t_2 = t_{OR} + t_{RQ} + t_{QD} \\ t_3 = t_{OP} + t_{PQ} + t_{QD} \\ Q = f_1 + f_2 + f_3 \end{cases}。$$

(二) 模型求解

在求解这个问题之前，回顾我们的假设：个体都是自私的，每一个人都只考虑自己的利益而非整个系统的利益。所以我们只要求解出整个系统的用户均衡解，因为只有这样的均衡条件下系统才可能保持稳定，任何偏离均衡解的解必然会导致系统的不稳定从而使得新的解向均衡解靠拢。系统的均衡解是 $t_1 = t_2 = \dots = t_k$ ，在这个均衡解下，系统中的个体没有激励选择别的路径，系统达到稳定状态。

1) 个体的均衡：没加入新道路的情况

在没有加入新道路的情况下，系统在 $t_1 = t_2$ 时达到均衡，即所有个体在对路径 1 和路径 2 的选择上没有偏好。如果不满足这个条件，在所用时间较长的路径上的个体就会有倾向改变自己的选择直到满足均衡条件，换言之，只有均衡条件下的解是稳定的。在这种情况下可以得到如下方程组

$$\begin{aligned} & t_1 = t_2 \\ & \text{st.} \begin{cases} f_1 + f_2 = Q \\ t_1 = a_{OR} + b_{OR}f_1 + a_{RS} + b_{RS}f_1 + a_{SD} + b_{SD}f_1 \\ t_2 = a_{OP} + b_{OP}f_2 + a_{PQ} + b_{PQ}f_2 + a_{QD} + b_{QD}f_2 \\ f_1, f_2, f_3 \geq 0 \end{cases} \end{aligned}$$

2) 个体的均衡：加入新道路之后的情况

到目前为止，我们已经表达出了在没有加入新道路之后的均衡时的方程，现在，我们考虑如果加入新的道路 (R, Q) 之后会发生什么，图 2 的右图表示了上面的状况。在加入了新的道路之后，这个交通网络被扩展了，在直觉上，人们希望加入新的道路会改善该路段的拥堵状况，即系统的总时间会降低。实际上是这个状况么？我们仍然利用上一小节中提到的寻找系统均衡的方法。

类似地，系统在 $t_1 = t_2 = t_3$ 时达到均衡，即所有个体在路段选择上没有偏好，系统达到稳定状态。在这种情况下满足如下的方程：

$$\begin{aligned} & t_1 = t_2 = t_3 \\ & \text{st.} \begin{cases} f_1 + f_2 + f_3 = Q \\ t_1 = a_{OR} + b_{OR}(f_1 + f_2) + a_{RS} + b_{RS}f_1 + a_{SD} + b_{SD}f_1 \\ t_2 = a_{OR} + b_{OR}(f_1 + f_2) + a_{RQ} + b_{RQ}f_2 + a_{QD} + b_{QD}(f_2 + f_3) \\ t_3 = a_{OP} + b_{OP}f_3 + a_{PQ} + b_{PQ}f_3 + a_{QD} + b_{QD}(f_2 + f_3) \\ f_1, f_2, f_3 \geq 0 \end{cases} \end{aligned}$$

在以上的两个情况中，我们只考虑 $f_k \geq 0$ 的情况，如果出现 $f_k < 0$ ，则认为这条路径上没有车辆经过，从而 $\sum_{i \neq k} f_i = Q$ 。对于没有加入新道路的情况，系统的总通行时间 $T_1 = f_1 t_1 + f_2 t_2$ ；在加入新的道路 (R, Q) 之后，系统的总通行时间为 $T_2 = f_1 t_1 + f_2 t_2 + f_3 t_3$ 。通过比较两者之间的大小得出车流量的范围从而确定路段的拥堵是否是由于 Braess 悖论引起的，当 $T_1 > T_2$ ，我们则认为新增道路缓解了整个交通网络的压力，Braess 悖论不存在；当 $T_1 < T_2$ 时，我们则认为新增道路使得整个交通网络更加拥堵，Braess 悖论存在。

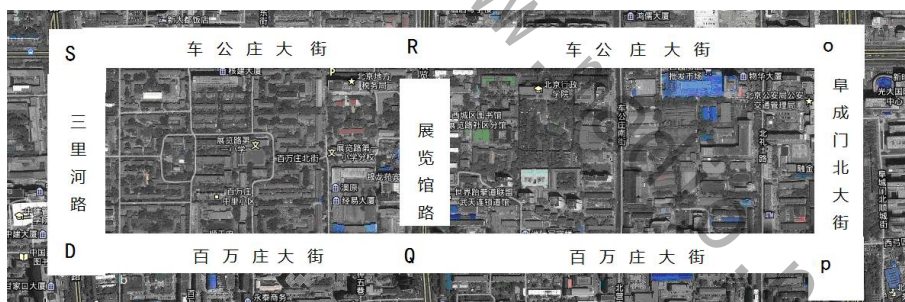
3) 实际问题的求解

分析连接北京二环路阜成门北大街的局域交通网络, 实际交通网络如下图 3, 该交通网络的数据来源于文献^[1], 数据见下表 1。同上面模型, 假设有 Q 辆车从 O 点进入该网络, 目的地为 D 点, 且同样只有 3 条路线可走, 路线 1 为 $ORSD$, 路线 2 为 $ORQD$, 路线 3 为 $OPQD$ 。

编号	长度(L) (米)	自由交通时间 a (秒)	延迟系数 b
1. 阜成门北大街	470	17.13	0.0163
2. 百万庄大街(右)	1036	27.32	0.0129
3. 百万庄大街(左)	844	21.31	0.0433
4. 三里河路	484	17.75	0.017
5. 车公庄大街(左)	892	24.90	0.0145
6. 车公庄大街(右)	1026	25.69	0.0245
7. 展览馆路	472	13.98	0.0078

1

表 1: 北京市某区域局部交通网络路段



2

图 3: 表 1 的数据所对应的北京市实际交通网络图

假设展览馆路即道路 RQ 为新增道路, 考虑有无道路 RQ 前后整个交通网络的变化, 利用上述模型, 代入实际数据, 可得到如下分析结果:

没有加入道路 RQ 前, 各路线的车流量 f_i 与总车流量 Q 如下:

$$\begin{cases} f_1 = 0.56Q - 20 \\ f_2 = 0.44Q + 20 \end{cases};$$

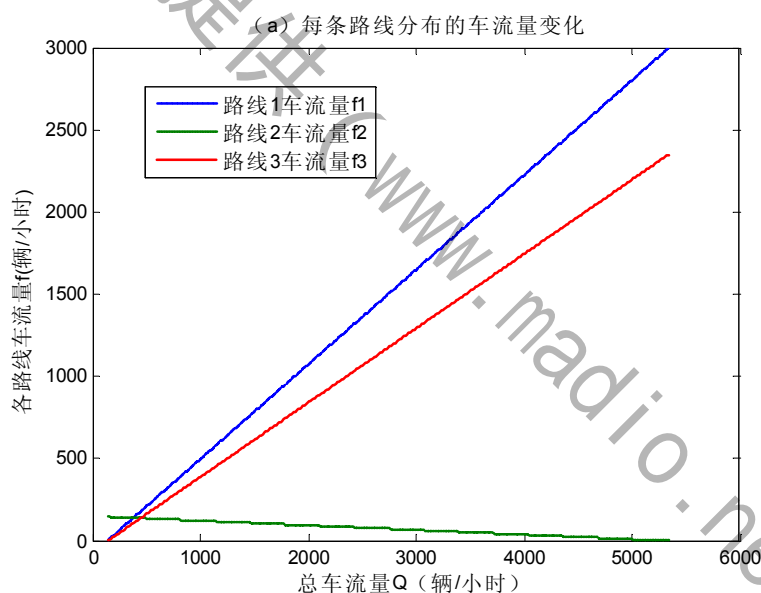
¹ 数据来源: 董菁等, 非合作交通网络中的 Braess 悖论及其避免, 公路交通科技, 21 卷第 5 期: 92-95, 2004..

² 图片来源: <http://ditu.google.cn/maps?hl=zh-CN&tab=wl>

加入道路 RQ 后，各路线的车流量 f_i 与总车流量 Q 如下：

$$\begin{cases} f_1 = 0.58Q - 83 \\ f_2 = 150 - 0.03Q \\ f_3 = 0.45Q - 67 \end{cases}$$

通过 Matlab 还可以求得在两种情况下系统的总通过时间之差 $T = T_1 - T_2$ 与总车流量 Q 的关系。另外，由加入道路 RQ 后各路线的车流量 f_i 与总车流量 Q 关系，我们可知当 $Q < 149$ 时， f_1 或者 f_3 小于 0；当 $Q > 5338$ 时， f_2 小于 0。所以在此我们先考虑 $149 \leq Q \leq 5338$ 时的情况，加入道路 RQ 后各路线的车流量 f_i 随总车流量 Q 的变化曲线图以及新增道路前后系统的总通过时间之差 T 随总车流量 Q 的变化曲线图如图 4 所示：



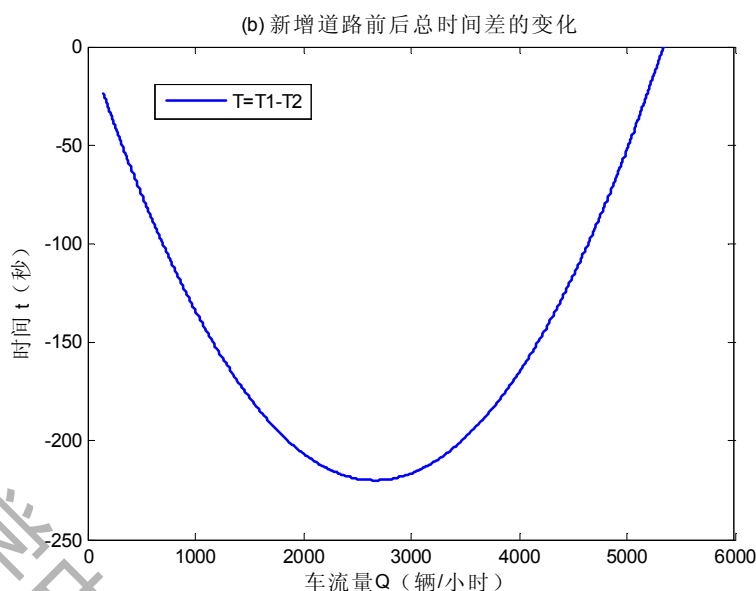


图 4: (a)图表示在加入新道路之后三条路线的车流量 f_i 随着车流量 Q 的变化趋势,其中路线 2 为新增路线;
(b)图表示新增道路前后系统总时间差 T 与车流量 Q 的关系;

通过图(a)可以看到新增道路的车流量很小,且随着总车流量 Q 的不断增加而减小,由此说明新增道路并没有很好的分流车辆,缓解交通。通过图(b)可以看到新增道路前后的系统总时间差小于 0,说明新增道路后系统不但没有变得畅通,反而更加拥堵, Braess 悖论存在。

当 $Q > 5338$ 时, f_2 小于 0, 我们所考虑的均衡情况不存在, 此时系统要达到最优时 f_2 等于 0, 即新增道路上没有车辆通行, 而原有路线 1, 3 上的车流量与没新增道路前一样, 因此系统总通过时间至少与没新增道路前系统相同, 可以说新增道路是浪费的, Braess 悖论存在。

当 $Q < 149$ 时, 由于总车流量 Q 太小, 实际中不大可能出现这种情况, 所以我们不予考虑。

(三) 结论

我们利用 2007 年 10 月 1 日上午北京二环路上的车流量和平均速度的数据统计了北京二环路阜成门北大街的车流量³, 数据描述的是, 以两分钟作为间隔观察通过某一截面的车流量以及车时速。我们假设当车时速小于 40km/h 时道路处于拥堵状态, 这样我们依据车速可以把数据分为两个时段, 一个时段表示交通处于通畅状态, 另一个时段表示交通处于拥堵状态。在我们对数据进行初步处理之后, 我们对二环路阜成门北大街的通畅时和拥堵时的车流量分别求算术平均, 得出在阜成门北大街通畅时和拥堵时每分钟通过截面的平均车辆数分别 36 辆和

³ 数据通过自行观测所得

93 辆。通过模型中 f_3 与 Q 的关系可以估计出 Q 在道路通畅时是 4926 辆/小时，在道路拥挤时是 12758 辆/小时，由前面的分析结果可知，这一部分交通网络存在 Braess 悖论。

4. 问题二模型的建立与求解

(一) 模型的建立

多主体模型由于其独具的优越性，近年来受到了不同领域的学者和专家的广泛关注。多主体模型较传统模型具有相当的优越性，一个主要的特点是多主体模型能够考虑到个体之间的异质性，并且方便讨论个体行为与整体表现行为之间的关系。鉴于此，我们引入多主体建模的方法对本问题进行讨论。

考虑到北京的实际交通网络情况，我们使用如图 5 所示的网格对北京的交通网络进行抽象。图中的每一个结点代表一个十字路口，假定从任何一个十字路口出发最多存在着 4 条线路，如果正好处在边界则仅仅存在两条可能的高速公路。为了体现现实中各条线路之间的异质性，我们假定在网络中存在如图 5 所示的两条高速环线。较一般路线而言，车辆在高速环线上行驶将会拥有更快的速度。当车辆在路口 S 时，其可能选择的路线只可能是 $S \rightarrow A$, $S \rightarrow B$, $S \rightarrow C$, $S \rightarrow D$ 中的一条。进一步我们假定，如果终点在 T 处，那么其可能的选择将会局限在 $S \rightarrow C$ 和 $S \rightarrow D$ 之中，也就是说我们在每一步的选择都应该使得我们更加靠近终点。

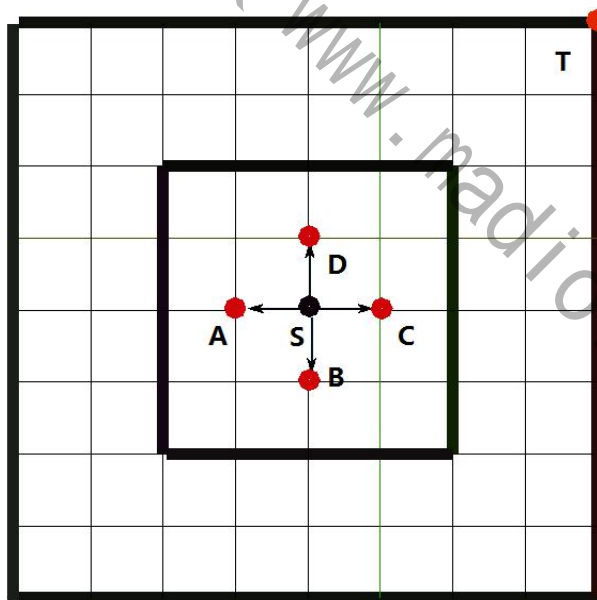


图 5：交通网络示意图

为了讨论司机广泛使用可以反映目前交通拥堵情况的 GPS 导航系统是否能够缓解交通阻塞，我们首先需要明确 GPS 导航系统能够提供怎样的导航信息。在模型中，我们假定使用 GPS 导航系统只能够提供给司机当前时刻当前路口和下一路口之间的车辆数。以图 5 为例，司机当前处于路口 S ，如果使用 GPS 导航，司机将会获得 $S \rightarrow A$, $S \rightarrow B$, $S \rightarrow C$, $S \rightarrow D$ 四条线路上的车辆数目。此假设

虽然与实际的 GPS 导航不完全符合，但是仍然具有相当的合理性。因为现实中 GPS 总是实时更新的，所以能够对决策起到主要作用的将是我们在假设中给出的局域信息。

在我们的模型中，我们将每一个车辆看成一个主体(Agent)，整个网络中存在 N_1 个主体，初始时刻随机地分布在网格中的结点上。网格中的结点分布均匀，并且每两个路口之间的距离为 1。道路 (i, j) 通过时间为 $t_{ij} = a_{ij} + b_{ij}f_{ij}$ ，其中， a 表示没有车辆阻塞的情况下通过该线路所需要的时间， b 表示受阻系数， f_{ij} 表示通过该道路的车流量。上式说明，位于任意路口处的车辆通过一条线路的时间与该条线路上车流量成正比。如果 $S = (x, y)$ 处的车辆的目的地是 $T = (x_0, y_0)$ ，那么引入 GPS 后的策略可以表示成为如下的模型：

$$X = \underset{A, B, C, D}{\text{Min}} \{t_{SA}, t_{SB}, t_{SC}, t_{SD}\}$$

$$st. \begin{cases} d_{XA} < d_{ST} \\ t_{ij} = a_{ij} + b_{ij}f_{ij} \end{cases}$$

其中， $d_{ST} = |x - x_0| + |y - y_0|$ 。

当不存在 GPS 进行导航的时候，我们不能够估计出任意线路上的车辆总数，因此也就不可能在路口处得到通过该线路实际所需要的时间。此种情况下的决策过程将可以表示为如下：

$$X = \underset{A, B, C, D}{\text{Min}} \{\bar{t}_{SA}, \bar{t}_{SB}, \bar{t}_{SC}, \bar{t}_{SD}\}$$

$$st. \begin{cases} d_{XA} < d_{ST} \\ \bar{t}_{ij} = a_{ij} \end{cases}$$

(二) 模型的初始条件与仿真

我们设定网络规模为 $N_0 \times N_0$ 的网格，整个网络中存在的车辆数目为 N_1 。网络中存在如图 5 所示的两条高速环线，给定在该高速环线上 $a_H = 0.5$ ，其他线路上 $a_L = 2$ ，并且所有的线路上都有 $b = 0.02$ 。初始时刻将 N_1 辆车随机分布在网络的格点上，并且令每一车辆的终点坐标为 (N_0, N_0) 。

首先考虑给定 $N_0 = 25$ ，随着网络中车辆数目 N_1 不断增加，整个交通网络的

运行效率的变化情况，模拟结果如图 6 所示。图 6 左面部分比较了存在 GPS 和没有 GPS 情况下所有车辆到达终点所需要的平均时间，其中红色线为存在 GPS 的情况，蓝色线为没有 GPS 的情况。不难看出，引入 GPS 是否有效以及在多大程度上有效于当前的交通网络中的车辆规模有关。如果现有的交通网络中车辆较少，引入 GPS 并不能够提高整个网络的运输效率，甚至可能会出现一定程度上适得其反的效果。我们认为出现此种情况的原因可能在于 GPS 导航系统使得交通系统中的司机做出了相同的决策，从而在一定的程度上表现出了预期自我毁灭的特征。当原有的运输系统承载量相对较小，车辆会比较均匀地分布在整个交通网络中。即使没有 GPS 导航系统，车辆也能够较快地到达终点。然而，当原有交通网络中承载的车辆达到一定程度之后，与没有 GPS 的情况相比，引入 GPS 将会显著改善整个交通网络的运行效率。进一步地，可以发现引入 GPS 后，所有车辆到达终点的平均时间与网络中承载的车辆数目呈比例增加，其上升速度显著低于不引入 GPS 的情况。图 6 的右面部分从另一个侧面反映了这一结论。当我们考察车辆在整个前进过程中的平均速度的时候，发现只有当车辆规模到达一定阈值的时，引入 GPS 后网络的表现才会显著优于没有 GPS 的情况。

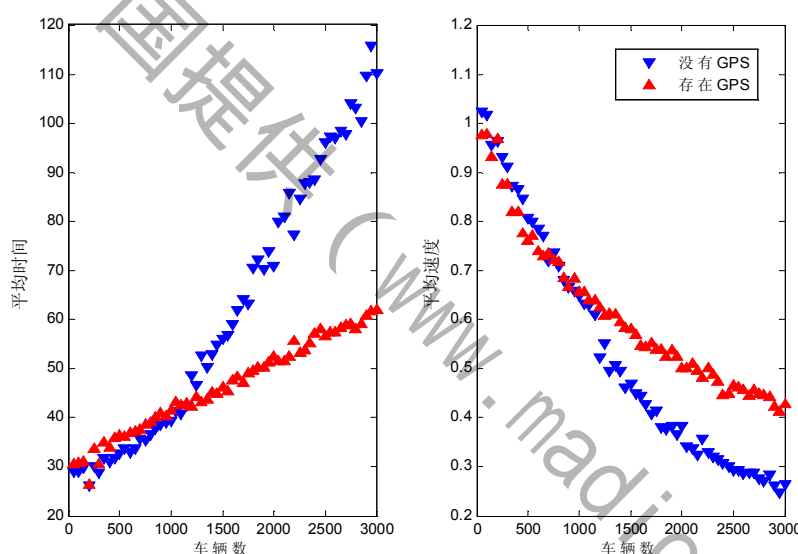


图 6: $N_0 = 25$ ，车辆数从 50 增加到 3000 时，整个交通网络的运行效率变化情况。

当我们需要改善交通状况的时候，一个自然的想法是扩建交通网络。因而我们需要考虑在扩大交通网络的同时，引入 GPS 是否仍然必需。取 $N_1 = 2000$ ，

考察 N_0 增加过程中整个网络的运行效率如何变化，模拟结果如图 7 所示。图 7 左面部分仍然表示所有的车辆到达终点所需平均时间的变化情况。由于我们在模拟的过程始终将终点固定在网络的最远端，车辆的出发点均匀地分布在整个网络节点上。这就意味着随着网络规模的不断增加，车辆的初始位置离终点的平均距离将会增加。因而随着 N_0 的增加，所需平均时间也将不断增加。但是在此过程中我们仍然可以发现引入 GPS 能够改善整个网络的运行效率。图 7 右面部分显示了随着 N_0 的变化，平均速度的变化情况。一个有意思的结果：如果没有 GPS

的导航，在网络规模增加的最开始反而会造成更大交通阻塞，此种情况正是 Braess 悖论的发生所致。因为网络规模的增加，将会使得车辆到达终点的路径增加，也就是为车辆提供了更加丰富的选择，由于此时没有 GPS 信息的导航，作为单个车辆并不了解网络的全局信息，并且个体都是自私的，他们只考虑自己的状况，从而造成整个网络更低的运行效率。然而，当网络规模不断增加，使得增加的路径充分多的时候，发生 Braess 悖论的可能性将会消失。此时，整个交通网络承载的车辆相对于网络本身而言显得很微小，因而车辆的平均速度能够较大幅度的提升。另外，我们考察引入 GPS 后的网络的运行效率如何变化。我们发现，引入 GPS 后将不会出现 Braess 悖论。平均速度随着网络规模的增加出现不断上升的趋势。这一模拟结果，为我们提出了解决 Braess 悖论的两种思路：一种简单的方法是引入 GPS，为司机提供相关的交通信息；另外一种思路是，扩建足够多的道路，从而降低 Braess 悖论发生的可能性。相比较而言，显然前者更经济可行。

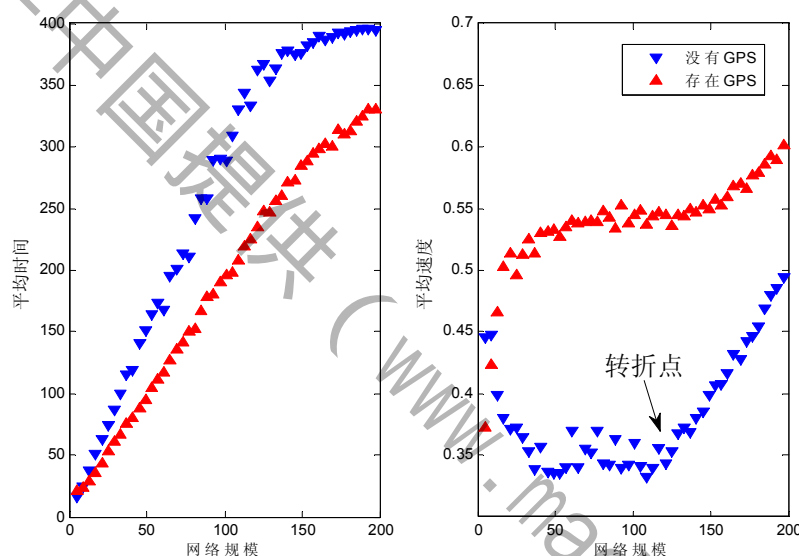


图 7： $N_1 = 2000$ ， N_0 从 5 增加到 201 时，整个交通网络运行效率变化情况。

在讨论整个交通网络的运行效率时，所有车辆的最后到达时间也是一个十分重要的指标。这一指标的深层意义正好体现了协同学中的役使原则，也就是所谓的慢变量决定快变量的过程。具体的模拟结果如图 8 所示。我们可以发现所有的车辆的最后到达时刻随着整个网络负荷量的增加而增加，并且引入 GPS 后的最后到达时间比没有 GPS 需要的时间更短。与图 6 得到的结论类似，引入 GPS 只有在原有的网络规模到达一定阈值后才能够起到显著改善作用，并且一旦超过此阈值使用 GPS 将会收到显著的改善效果。

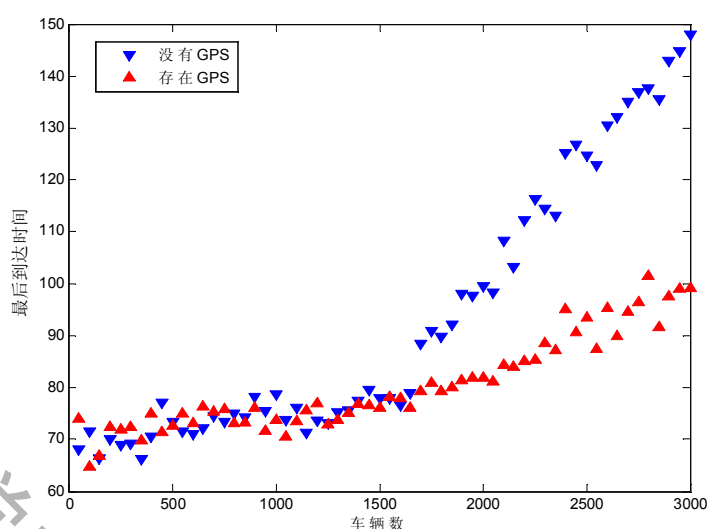


图 8：最后到达时刻随网络承载量的变化

为了改善局部交通能力而增加投入最终却使得整体的交通状况更加恶化，这样将会造成巨大的浪费，这也正是 Braess 悖论引起广泛关注的原因。因此，为了节约成本我们总是希望在承载同样的车流量条件下，网络中连边的数量越少越好。既然广泛使用 GPS 之后能够提高整个运输过程的效率，那么是否在提高运输效率的同时也能够更节约成本呢？针对这一问题，我们使用同样的模型对此进行了初步地分析。考虑随着网络承载量的 N_1 不断增加，我们现有的交通网络是否得到了充分地利用。从图 9 我们可以看出，使用 GPS 之后网络的公路空置率始终高于不使用 GPS 的情况，也就是说使用 GPS 能够以更小规模的网络承载相同的车流量。综合前面得出的结论我们可以知道，在一定量的车流量下，使用 GPS 将会显著提高运输效率和节约成本。

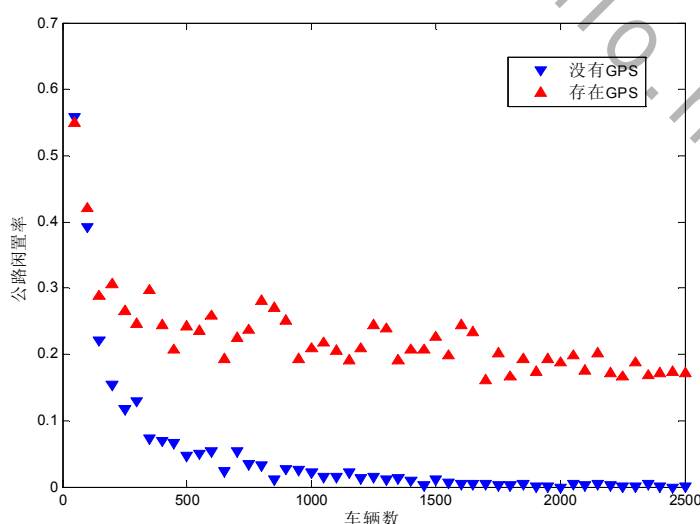


图 9： $N_0 = 25$ ，公路空置率随 N_1 的变化

(三) 模型参数分析

本文仅仅选取一个有代表性的重要参数 a_H 进行讨论，此参数衡量了修建的高速环线上的运行速度。由于模型初始给定的一般线路上的相应参数为 $a_L = 2$ ，因此我们在区间 $[0.5, 2]$ 上对 a_H 进行讨论。我们选取网络规模 $N_1 = 2500$, $N_0 = 25$ 。相应的模拟结果如下图所示。当 $a_H = 0.5$ 时，等价于我们前文中设定的模型参数，显然此条件下的模拟结果和前文中结果一致。 a_H 不断增加，意味着此时的高速环线本身相对于普通线路的优势逐渐降低。模拟结果显示，随着 a_H 不断增加，整个网络的运行效率在逐渐下降；另外可以看出没有 GPS 和存在 GPS 两种情况下的效率下降速度几乎相同。换一个角度观察，我们发现当 $a_H = 0.5$ 没有 GPS 时的平均效率远远低于 $a_L = 2$ 存在 GPS 的情况下的平均效率。这一事实说明在当前设定的网络状态下，修建高速环线对网络的改善效果远不如广泛使用 GPS 带来的改善效果。

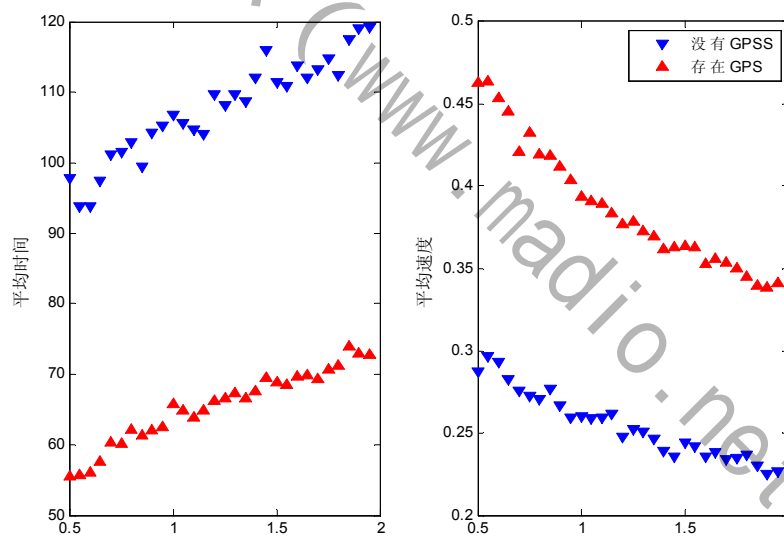


图 10: 参数 a_H 对网络运行效率的影响

(四) 结论

通过以上的多主体模型，我们知道如果当前的交通网络满足一定条件后广泛使用 GPS 导航系统确实能够显著缓解整个交通网络的拥挤程度。具体到北京市的交通情况，需要根据实际测算的交通网络规模，车辆数量，以及道路好坏情况进行估计。一个基本的结论是，如果当前网络中的车辆总数与整个网络的规模相比较小的时候，使用 GPS 会适得其反。我们构建的模型的一个重要特点是能够再现

经典的 Braess 悖论，如图 7 右图。在特定的参数条件下，如果扩建网络反而会降低整个交通网络的运行效率。因此，只要根据实际情况合理估计参数，我们提出的多主体模型可以检验现有交通网络中的某些堵塞拥挤情况是否来源于 Braess 悖论。

与此同时，当网络中的车辆总数超过测算阈值时，引入 GPS 能够在较大程度上缓解整个网络的交通状况。进一步，我们还发现引入 GPS 在提高整个网络运行效率的同时，更加节约成本。

5. 模型总体的结论及政策建议

通过以上对模型一的求解可以得出在北京市二环路(包括二环路)以内，确实存在 Braess 悖论，展览馆路就是这样一个例子，这个事实提醒我们，在考虑优化交通网络系统时要慎重考虑道路参数对实际情况的影响。通过对模型二的求解，我们得出合理使用 GPS 的确能够帮助我们优化当前存在的网络，引导交通流，进而区别整个网络中的高效线路和低效线路。这将有利于相关部门对整个网络进行科学管理，将紧缺的维护资源重点放在高效线路上。另外，可以在我们模型的基础上进行扩展，讨论在有限预算的前提下，采用何种方式对现有网络进行改建能够最大程度地提高现有交通网络运行效率，这一扩展将具有很强的现实意义。

参考文献：

- [1]http://www.sina.com.cn, 北京晚报, 2009 年 12 月 15 日.
- [2]陈彦光, 刘级生, Braess 模型与城市网络的空间复杂化探讨, 地理科学, 第 26 卷第 6 期: 659-663, 2006.
- [3]Wilson A G, Complex Spatial Systems: The Modeling Foundations of Urban and Regional Analysis, Singapore: Pearson Education Asia Pte Ltd, 2000.
- [4]Braess D, A Paradox on Traffic Networks, <http://homepage.ruhr-uni-bochum.de/>, 2004.
- [5]Arnott R, Small K, The Economics of Traffic Congestion, American Scientist, Vol.82:446-455, 1994.
- [6]Lam W H K, Effects of Road Pricing on System Performance, Traffic Eng. Control, Vol.29, No.12:631-635:1988.
- [7]Pas E, Principio S, Braess'paradox: Some New Insights, Transpn Res.-B, Vol.31, No.3:265-276, 1997.
- [8]董菁, 张佐, 非合作交通网络中的 Braess 悖论及其避免, 公路交通科技, 21 卷第 5 期: 92-95, 2004.

附录：

● main.m 文件的代码

```
NT=30;
averagetime=zeros(1,NT);
averagevelocity=zeros(1,NT);
emptyhighway=zeros(1,NT);
emptyhighwaywithgps=zeros(1,NT);
averagetimewithgps=zeros(1,NT);
averagevelocitywithgps=zeros(1,NT);
for i=1:NT
    i
    [meannogps, longesttimenogps, averagevelocitynogps, edgex, edgey]=meantimenogps(i);
    averagetime(i)=meannogps;
    averagevelocity(i)=averagevelocitynogps;
    emptyhighway(i)=length(find(edgex==0))+length(find(edgey==0));

    [meanwithgps, averagewithgps, longesttimewithgps, edgexwithgps, edgeywithgps]=meantimewithgps(i);
    averagetimewithgps(i)=meanwithgps;
    averagevelocitywithgps(i)=averagewithgps;
    emptyhighwaywithgps(i)=length(find(edgexwithgps==0))+length(find(edgeywithgps==0));
end
% t0=1:NT;
% N1=50.*t0;
t0=1:NT;
N1=2-0.05*t0;
subplot(1,2,1)
plot(N1,averagetime)
hold on
plot(N1,averagetimewithgps,'r')
hold off
subplot(1,2,2)
plot(N1,averagevelocity)
hold on
plot(N1,averagevelocitywithgps,'r')
hold off
% plot(N1,emptyhighway)
% hold on
% plot(N1,emptyhighwaywithgps,'r')
% hold off
```

● meantimenogps.m 文件的代码：

```
Function [meannogps, longesttimenogps, averagevelocitynogps, edgex, edgey]=meantimenogps (NTIME)
N1=2500;    %number of cars in the transport network
alpha=2-0.05*NTIME;
global N0
N0=25;
global SOURCEX
global SOURCEY
sourcecx=zeros(1,N0);
sourcecy=zeros(1,N0);
edgex=zeros(N0-1,N0);
edgey=zeros(N0,N0-1);
targetx=N0;
targety=N0;
global DISTANCE
distance=zeros(1,N1); %离目标点的当前距离
dis=zeros(1,N1); %行进过程中离目标点的距离
choice=zeros(1,N1);
velocity=zeros(1,N1); % the velocity of each agent
time=zeros(1,N1); % 记录每一车辆用时
D=zeros(1,4);
global corrx
global corry
corrx=zeros(N0,N0);
corry=zeros(N0,N0);
for i=1:N0
    for j=1:N0
        if rem(i,(N0-1)/4)==1 || rem(i,N0-1)==0 || rem(j,(N0-1)/4)==1 || rem(j,(N0-1))==0
            corrx(i,j)=alpha; % 没有交通阻塞时候所需要的时间 横向边
            corry(i,j)=alpha; %没有交通阻塞时候所需要的时间 纵向边
        else
            corrx(i,j)=2; % 没有交通阻塞时候所需要的时间 横向边
            corry(i,j)=2; %没有交通阻塞时候所需要的时间 纵向边
        end
    end
end
global b
b=0.02;
% 给定初始时刻agent的出发点和终点
for k=1:N1
    sourcecx(k)=randint(1,1,[1,N0-1]); %random choose the source of agent i
    sourcecy(k)=randint(1,1,[1,N0-1]);
    distance(k)=abs(sourcecx(k)-targetx)+abs(sourcecy(k)-targety);
end
```

```

end
SOURCEX=sourcecx;
SOURCEY=sourcecy;
DISTANCE=distance;
% 个体在没有GPS导航的情况之下作出选择,事实上在网格中每次选择只需要作出方向的判定即可,1:上 2:
下 3:左 4:右;
NT=0; %表示的是已经到达终点的车辆数量
T=400; % upperbond of time
crit=0.05;
step=0.02; % time step, for we could not have the continuous time
for t1=0:0.02:T % time step 0.02
if NT<N1 %首先判断网络中是否还有车辆
for i=1:N1
dis(i)=abs(sourcecx(i)-targetx)+abs(sourcecy(i)-targety); %
% whether the current position of agent i is close to the stop if so, we
% dicide which way to choose next.
if abs(ceil(sourcecx(i))-sourcecx(i))<crit&&abs(ceil(sourcecy(i))-sourcecy(i))<crit
sourcecx(i)=round(sourcecx(i));
sourcecy(i)=round(sourcecy(i));
dis(i)=abs(sourcecx(i)-targetx)+abs(sourcecy(i)-targety); %当前点到目标的距离
if dis(i) >0
D(1)=abs(sourcecx(i)-targetx)+abs(sourcecy(i)+1-targety); %往上走
if sourcecy(i)+1<=N0
D(1)=D(1);
else D(1)=100000; %惩罚项
end
D(2)=abs(sourcecx(i)-targetx)+abs(sourcecy(i)-1-targety); %往下走
if sourcecy(i)-1>=1
D(2)=D(2);
else D(2)=100000;
end
D(3)=abs(sourcecx(i)-1-targetx)+abs(sourcecy(i)-targety); %往左走
if sourcecx(i)-1>=1
D(3)=D(3);
else D(3)=100000;
end
D(4)=abs(sourcecx(i)+1-targetx)+abs(sourcecy(i)-targety); %往右走
if sourcecx(i)+1<=N0
D(4)=D(4);
else D(4)=100000;
end
D1=D-dis(i);
index=find(D1<0);
%可能前进的方向,需要进一步判断走哪一个具体方向,根据预期的速度来进行判断

```

```

L=length(index);
t=zeros(1,L);
tn=zeros(1,L);
v=zeros(1,L);
for j=1:L
    tempt1=index(j);
    % GPS is not allowed here, so that when make decisions we do not know exactly the value of
    NUM and we assume NUM is zero
    % However, when we caculate the realized required time, we should count NUM
    % one thing is that, we should notice the time when cars arrive the next station
    switch tempt1
        case 1
            A= sourcec<=sourcec(i)+1;
            C=find(sourcec(A)==sourcec(i));
            NUM=length(C);
            if sourcec(i)==N0-1
                NUM=0;
            end
            a=corry(sourcec(i),sourcec(i));
            tn(j)=a;
            t(j)=a+b*NUM;
            v(j)=1./t(j);
        case 2
            A=find(sourcec>=sourcec(i)-1);
            C=find(sourcec(A)==sourcec(i));
            NUM=length(C);
            a=corry(sourcec(i),sourcec(i)-1);
            tn(j)=a;
            t(j)=a+b*NUM;
            v(j)=1./t(j);
        case 3
            A=find(sourcec>=sourcec(i)-1);
            C=find(sourcec(A)==sourcec(i));
            NUM=length(C);
            a=corrx(sourcec(i)-1,sourcec(i));
            tn(j)=a;
            t(j)=a+b*NUM;
            v(j)=1./t(j);
        case 4
            A=find(sourcec<=sourcec(i)+1);
            C=find(sourcec(A)==sourcec(i));
            NUM=length(C);
            if sourcec(i)==N0-1
                NUM=0;
            end
    end
end

```

```

        end
        a=corr(x(sourcex(i), sourcey(i)));
        tn(j)=a;
        t(j)=a+b*NUM;
        v(j)=1./t(j);
    end
end
index1=find(tn-min(tn)==0); % make choice without gps
index1=index1(randint(1,1,[1,length(index1)]));
choice(i)=index(index1);
tbar=t(index1); % real time required the same as the case with gps
velocity(i)=v(index1); % agent have different velocity
% update the current position according to the choice
tempt=choice(i);
switch tempt
case 1
    %record which highway agent i enter, and update the number of
    %cars passed that high way
    edgex(sourcex(i), sourcey(i))= edgex(sourcex(i), sourcey(i))+1;
    sourcex(i)=sourcex(i);
    sourcey(i)=sourcey(i)+velocity(i)*step;
case 2
    edgex(sourcex(i), sourcey(i))= edgex(sourcex(i), sourcey(i)-1)+1;
    sourcex(i)=sourcex(i);
    sourcey(i)=sourcey(i)-velocity(i)*step;
case 3
    edgex(sourcex(i), sourcey(i))= edgex(sourcex(i)-1, sourcey(i))+1;
    sourcey(i)=sourcey(i);
    sourcex(i)=sourcex(i)-velocity(i)*step;
case 4
    edgex(sourcex(i), sourcey(i))= edgex(sourcex(i), sourcey(i))+1;
    sourcey(i)=sourcey(i);
    sourcex(i)=sourcex(i)+velocity(i)*step;
end
time(i)=time(i)+tbar;
end
else
    tempt3=choice(i);
    switch tempt3
    case 1
        sourcey(i)=sourcey(i)+velocity(i)*step;
        sourcex(i)=sourcex(i);
    case 2

```

```

        sourcey(i)=sourcey(i)-velocity(i)*step;
        sourcex(i)=sourcex(i);
        case 3
            sourcey(i)=sourcey(i);
            sourcex(i)=sourcex(i)-velocity(i)*step;
        case 4
            sourcey(i)=sourcey(i);
            sourcex(i)=sourcex(i)+velocity(i)*step;
        end
    end
end
else
    break %if all the cars have arrived to tha target then break. stop running the program.
end
NT=length(find(dis==0));
end
meannogps=mean(time);
longesttimenogps=max(time);
averagevelocitynogps=mean(distance./time);

```

● meantimewithgps.m 的代码

```

function
[meanwithgps, averagewithgps, longesttimewithgps, edgexwithgps, edgeywithgps]=meantimewithgps(NTIME)
N1=2500;
alpha=2-0.05*NTIME;
global N0
global SOURCEX
sourcex=SOURCEX;
global SOURCEY
sourcey=SOURCEY;
edgexwithgps=zeros(N0-1,N0); %记录每一条横向边的通行状况
edgeywithgps=zeros(N0,N0-1); %记录每一条纵向边的通行状况
global corrx
global corry
global b
targetx=N0;
targety=N0;
global DISTANCE
distancewithgps=DISTANCE;
dis=zeros(1,N1); %行进过程中离目标点的距离
choice=zeros(1,N1);
velocity=zeros(1,N1); % the velocity of each agent

```

```

timewithgps=zeros(1,N1); % 记录每一车辆用时
D=zeros(1,4);
% 个体在没有GPS导航的情况之下作出选择,事实上在网格中每次选择只需要作出方向的判定即可,1:上 2:
下 3:左 4:右;
NT=0; %表示的是已经到达终点的车辆数量
T=400; % upperbond of time
crit=0.05;
step=0.02; % time step, for we could not have the continious time
for t1=0:0.02:T % time step 0.02
if NT<N1 %首先判断网络中是否还有车辆
for i=1:N1
dis(i)=abs(sourcex(i)-targetx)+abs(sourcey(i)-targety); %
% whether the current position of agent i is close to the stop if so, we
% dicide which way to choose next.
if abs(ceil(sourcex(i))-sourcex(i))<crit&&abs(ceil(sourcey(i))-sourcey(i))<crit
sourcex(i)=round(sourcex(i));
sourcey(i)=round(sourcey(i));
dis(i)=abs(sourcex(i)-targetx)+abs(sourcey(i)-targety); %当前点到目标的距离
if dis(i) >0
D(1)=abs(sourcex(i)-targetx)+abs(sourcey(i)+1-targety); %往上走
if sourcey(i)+1<=N0
D(1)=D(1);
else D(1)=100000; %惩罚项,使得agent不会选择走出边界
end
D(2)=abs(sourcex(i)-targetx)+abs(sourcey(i)-1-targety); %往下走
if sourcey(i)-1>=1
D(2)=D(2);
else D(2)=100000;
end
D(3)=abs(sourcex(i)-1-targetx)+abs(sourcey(i)-targety); %往左走
if sourcex(i)-1>=1
D(3)=D(3);
else D(3)=100000;
end
D(4)=abs(sourcex(i)+1-targetx)+abs(sourcey(i)-targety); %往右走
if sourcex(i)+1<=N0
D(4)=D(4);
else D(4)=100000;
end
D1=D-dis(i);
index=find(D1<0);
%可能前进的方向,需要进一步判断走哪一个具体方向,根据预期的速度来进行判断
L=length(index);
t=zeros(1,L);

```

```

tn=zeros(1,L);
v=zeros(1,L);
for j=1:L
temptl=index(j);
    % GPS is not allowed here, so that when make decisions we do not know exactly the value of
NUM and we assume NUM is zero
    % However, when we caculate the realized required time, we should count NUM
    % one thing is that, we should notice the time when cars arrive the next station
switch temptl
    case 1
        A=find(sourcey<=sourcey(i)+1);
        C=find(sourcex(A)==sourcex(i));
        NUM=length(C);
        if sourcey(i)==N0-1
            NUM=0;
        end
        a=corry(sourcex(i), sourcey(i));
        tn(j)=a;
        t(j)=a+b*NUM;
        v(j)=1./t(j);
    case 2
        A=find(sourcey>=sourcey(i)-1);
        C=find(sourcex(A)==sourcex(i));
        NUM=length(C);
        a=corry(sourcex(i), sourcey(i)-1);
        tn(j)=a;
        t(j)=a+b*NUM;
        v(j)=1./t(j);
    case 3
        A=find(sourcex>=sourcex(i)-1);
        C=find(sourcey(A)==sourcey(i));
        NUM=length(C);
        a=corr(x(sourcex(i)-1, sourcey(i));
        tn(j)=a;
        t(j)=a+b*NUM;
        v(j)=1./t(j);
    case 4
        A=find(sourcex<=sourcex(i)+1);
        C=find(sourcey(A)==sourcey(i));
        NUM=length(C);
        if sourcex(i)==N0-1
            NUM=0;
        end
        a=corr(x(sourcex(i), sourcey(i));

```

```

        tn(j)=a;
        t(j)=a+b*NUM;
        v(j)=1./t(j);
    end
end
index1=find(t-min(t)==0); % make choice without gps
index1=index1(randint(1,1,[1,length(index1)]));
choice(i)=index(index1);
tbar=t(index1);          % real time required   the same as the case with gps
velocity(i)=v(index1);    % agent have different velocity
% update the current position according to the choice
tempt=choice(i);
switch tempt
    case 1
        %record which highway agent i enter, and update the number of
        %cars passed that high way
        edgewithgps(sourcecx(i),sourcecy(i))= edgewithgps(sourcecx(i),sourcecy(i))+1;
        sourcecx(i)=sourcecx(i);
        sourcecy(i)=sourcecy(i)+velocity(i)*step;
    case 2
        edgewithgps(sourcecx(i),sourcecy(i))= edgewithgps(sourcecx(i),sourcecy(i)-1)+1;
        sourcecx(i)=sourcecx(i);
        sourcecy(i)=sourcecy(i)-velocity(i)*step;
    case 3
        edgewithgps(sourcecx(i),sourcecy(i))= edgewithgps(sourcecx(i)-1,sourcecy(i))+1;
        sourcecy(i)=sourcecy(i);
        sourcecx(i)=sourcecx(i)-velocity(i)*step;
    case 4
        edgewithgps(sourcecx(i),sourcecy(i))= edgewithgps(sourcecx(i),sourcecy(i))+1;
        sourcecy(i)=sourcecy(i);
        sourcecx(i)=sourcecx(i)+velocity(i)*step;
end
timewithgps(i)=timewithgps(i)+tbar;
end
else
    tempt3=choice(i);
    switch tempt3
        case 1
            sourcecy(i)=sourcecy(i)+velocity(i)*step;
            sourcecx(i)=sourcecx(i);
        case 2
            sourcecy(i)=sourcecy(i)-velocity(i)*step;
            sourcecx(i)=sourcecx(i);
        case 3

```

```
        sourcey(i)=sourcey(i);
        sourcex(i)=sourcex(i)-velocity(i)*step;
    case 4
        sourcey(i)=sourcey(i);
        sourcex(i)=sourcex(i)+velocity(i)*step;
    end
end
end
end
else
    break    %if all the cars have arrived to the target then break. stop running the program.
end
NT=length(find(dis==0));
end
meanwithgps=mean(timewithgps);
longestimewithgps=max(timewithgps);
averagewithgps=mean(distancewithgps./timewithgps);
```