

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

网址：www.tzmcm.cnEmail：2010@tzmcm.cn

邮编：010021

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：林海卓

队员 2：杨哲

队员 3：占海明

参赛队教练员（签名）：王震

参赛队伍组别：大学本科组 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

网址：www.tzmcm.cn

Email：2010@tzmcm.cn

邮编：010021

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）：
1013

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.cnEmail：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 基于 Nash 均衡原理、小波神经网络模型及 ATM 动态导航路由选择对北京二环内 BRAESS 悖论情况的研究

关 键 词 Nash 平衡点 Pareto 边界 小波神经网络 Nash 平衡比率 路段延迟系数 ATM 路由选择 Hopfield 双重选路算法 灰色 Gompertz 预测 遗传算法

摘 要：

本文首先通过对非合作网络中 Nash 平衡点与 Pareto 边界理论的分析，通过单起点单终点的简单路网入手，分析 Braess 悖论的成因，Braess 悖论实质上是非合作网络中 Nash 平衡点不满足 Pareto 最优性时出现的现象。定量的给出出行时间与流量的关系，这里引入了路段延迟参数的概念，量化得到延迟参数与路段流量，出行时间，出行密度之间的关系。进而通过建立北京二环区交通路网模型，拟合出 $flow \sim L / speed$ 关系曲线，得到各路段的延迟参数，进而得到单月内北京二环区 Pareto 最优解的判别条件。

其次，根据 Pareto 边界理论，我们以年月为单位统计北京二环以内交通流量因素，得到城区交通年延迟参数，并计算了 2000-2009 年共 120 个月的延迟参数。接着，我们构建了 Nash 均衡原理模型，引入了路段 Nash 平衡比率的概念，即当某一时间范围内，路段 Nash 平衡比率低于路段延迟参数时，则发生了 Braess 悖论，当同时又满足 Nash 平衡波动率超过 30% 时，为严重 Braess 悖论，从而完善了 Nash 平衡比率理论方程。

再次，根据已建立的 Nash 平衡比率理论方程，在此基础上构建小波神经网络来对北京二环区交通路段 Nash 平衡比率进行预测，这里我们将选择 2000~2009 年路段延迟参数为样本，根据路段延迟参数与路段 Nash 平衡比率的动态关系式，对 2000~2009 年路段 Nash 平衡比率进行计算。得到结果：东城区，朝阳区路段常年持续存在 BRAESS 悖论情况（平均每年超过 6 个月）。尽管我们发现 2002 年（6 月天坛路拓宽），2003 年（6 月普方路拓宽），2004 年（7 月永定门桥延修），2008 年（9 月天坛路改道）会给北京二环区南部路段带来交通改善，但结果显示，整个北京二环路段的交通情况的平衡性变得更不稳定，BRAESS 悖论情况更加突出。譬如在 2006 年，2009 年的东城区，朝阳区路段堵塞问题更加尖锐，这也使得本来相对通畅的南段也出现 BRAESS 悖论情况。

接着，我们建立了 GPS 动态导航的车辆运行模型，量化得到了时耗期望值和时耗方差两个目标函数并作为 ATM 路由选择的约束条件，以降低 Nash 平衡比率为目标，通过 ATM 路由选择进行 GPS 导航对路段影响的预测。当我们控制业务源提高 GPS 使用率时，路段 NASH 平衡比率也持续上升，使用率升至 0.34 时，NASH 平衡比率上升幅度最大化。

最后，我们通过灰色 Gompertz 模型对 ATM 路由选择算法进行误差分析，检验结果表明此抽检方案为合理方案，误差范围 5.84%。通过遗传算法对路段 Nash 平衡比率进行优化，得到最优化后的结果，证明了 ATM 路由选择算法对 GPS 动态导航研究的可靠性。

参赛队号 1013

所选题目 B 题

参赛密码 _____
(由组委会填写)

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.cnEmail：2010@tzmcm.cn

英文摘要（选填）

（此摘要非论文必须部分，选填可加分，加分不超过论文总分的 5%）

Abstract

Firstly, through Nash equilibrium in the non-cooperative network and Pareto analysis of the boundary theory, we start from a network with a single starting point and a single ending point to analysis the reasons of Braess Paradox. Braess Paradox appears when Nash equilibrium in the non-cooperative network doesn't satisfy Pareto optimality. In order to give a quantitative relationship between travel time and traffic, we introduce in the concept of delay parameter to get the relation between delay parameter and link flows、traffic time、traffic density, meanwhile, get the delay parameters of each link, so we obtain the criterion of Pareto optimality in a single month.

Secondly, according to Pareto analysis of the boundary theory, we statistic Beijing traffic factors within the Second Ring, get delay parameter in urban traffic in one year and calculate the delay parameter from 2000 to 2009. Then, we construct Nash equilibrium theory model, introduce in the concept of Nash equilibrium ratio.

Thirdly, on the base of Nash equilibrium ratio theory equation, wavelet neural network model is constructed to predict the Nash equilibrium ratio the Second Ring. Dentally, we take the delay parameter from 2000 to 2009 for instance to calculate the Nash equilibrium ratio and draw the conclusion.

Then, GPS vehicle navigation operation dynamic model is constructed to calculate the time-consuming expectation objective function and the time-consuming variance objective function, and we take them as the constraint conditions of ATM routing choice to decrease the impact of GPS vehicle navigation on road section.

Finally, we analysis the errors of ATM routing choice through Gray Gompertz Model and the result shows the sampling inspection plan is reasonable. Meanwhile, the Genetic Algorithm optimization outcomes of Nash equilibrium ratio prove ATM routing choice algorithm is reliable.

目 录

二、问题分析	2
三、模型假设与符号说明	2
3.1 模型假设	2
3.2 符号说明	2
四、非合作网络中 NASH 平衡点与 PARETO 边界	3
4.1 BRAESS 悖论的出现本质	3
4.2 BRAESS 悖论的出现	3
4.3 出行时间与流量关系	4
4.4 北京二环区交通路网中单月的 BRAESS 悖论	5
4.5 2000-2009 年北京二环区交通路网中的 BRAESS 悖论	7
五、NASH 均衡原理与小波神经网络预测模型	9
5.1 NASH 均衡原理	9
5.2 构建小波神经网络对路段 NASH 平衡比率进行预测	10
5.2.1 模型的构建	10
5.2.2 小波函数的确定	11
5.2.3 对隐层小波元的确定	12
5.2.4 模型的求解	12
5.3 结论一：近十年北京二环路段 BRAESS 悖论情况	15
5.4 结论二：NASH 平衡比率与路段延迟系数的模拟结果	16
六、基于 GPS 动态导航的车辆运行模型	20
6.1 GPS 交通系统的动态随机特性	20
6.2 车辆导航用户	20
6.3 动态随机特性的数学描述	20
6.4 动态随机状态下的路径寻优	22
七、基于 ATM 路由选择对 GPS 动态导航对路段影响的预测	23
7.1 ATM 路由选择	23
7.2 用人工神经网络方法求解 ATM 路由选择问题	24
7.3 HOPFIELD 网络的双重选路法	25
7.4 算法模拟预测结果：	26
八、基于灰色 GOMPERTZ 模型对算法误差的分析	30
九、基于遗传算法对模型的优化检验	32
9.1 遗传算法流程图	32
9.2 遗传算法的实现	33
9.2.1 目标函数	33
9.2.2 选择与交叉	33
9.2.3 变异	34
9.2.4 最优保存策略：	34
9.3 优化结果及遗传状态的分析	34
9.4 结论：	36
参考文献	37
附 录	38

基于 NASH 均衡原理、小波神经网络模型及 ATM 动态导航路由选择对北京二环内 BRAESS 悖论情况的研究

一、问题重述

BRAESS 悖论分析

1968 年意大利数学家 DIETRICH BRAESS 发现交通网络中的 BRAESS 悖论现象。在此之前，人们一直认为任意交通设施的改善能提高交通系统通行能力。然而 BRAESS 指出：不考虑网络出行需求和路径选择原则，单方面地增加路网中的路段可能会使路网的通行状况变差。BRAESS 就满足 WARDROP 第一出行原则的用户平衡分配问题给出了一个实例，即在一个交通网络上增加一条路段，使网络上的出行时间增加，而且是所有出行者的出行时间都增加。这一附加路段不但没有减少交通延误，反而降低了整个交通网络的服务水平，这种与人们预期相悖的交通网络现象就是人们所说的“BRAESS 悖论”。

二、问题分析

首先分析单一路段在单一时间范围内的 BRAESS 悖论情况，然后以北京二环以内的综合路段为考虑对象，时间范围 2000~2009 年，并主要提取出 2002 年（6 月天坛路拓宽），2003 年（6 月普方路拓宽），2004 年（7 月永定门桥延修），2008 年（9 月天坛路改道）的 NASH 平衡比率，并通过 MATLAB 工具 CHIP（）提取出月波动率超过 30% 的比率数进行标定，同时，我们将近十年的北京二环以内主要路段的 BRAESS 悖论情况进行分析，并讨论 GPS 动态导航对路段 BRAESS 悖论现象的影响。

三、模型假设与符号说明

3.1 模型假设

- (1) 天气因素对 BRAESS 悖论的影响忽略不计
- (2) 《北京交通发展年报》及《全市综合交通调查》提供了精确无误的数据
- (3) 北京市区内地铁路线改造的影响忽略不计。

3.2 符号说明

变量和缩略语	定义
t_{ij}	路段 ij 上的出行时间
a_{ij}	自由流情况下的出行时间
b_{ij}	路段 ij 上的延迟参数
f_{ij}	路段 ij 上的车流量
op	瓶颈路段

qr	瓶颈路段
$f(t)$	时间序列
$g(t)$	路段 NASH 平衡比率预测值序列
W_k	权系数
a_k	小波基伸缩因子
b_k	小波基平移因子
V_0	最优染色体
M	群体中个体总数
N	小波基个数

四、非合作网络中 NASH 平衡点与 PARETO 边界

4.1 BRAESS 悖论的出现本质

1968 年 DIETRICH BRAESS 在他的一篇论文中提出在平衡交通流网络中存在一种看似矛盾的现象，即某种情况下，在交通网络中增加一条通路反而会增加网络中所有用户的出行时间。这一违背常理的现象被称为 BRAESS 悖论。ARNOTT 和 SMALL[1] (1994), BASS[2] (1992), CALVERT 和 KEADY (1993), YANG 和 BELL[3] (1998), DAGANZO[4] (1998) 等许多人在交通以及更广泛的领域内对该现象进行了研究与讨论。BRAESS 悖论的出现本质是由于非合作网络中的 NASH 平衡点不在 PARETO 边界上，这种情况下，存在一种非平衡的流量分布，使网络相对于平衡流量分布时某些用户的出行时间缩短，同时其他用户的出行时间也不会增加。

随着近年来城市路网中交通拥挤现象在时间和空间范围上的不断扩大，以及智能交通系统(ITS)研究的发展，人们在新路网的规划设计和路网的管理控制中更加深刻的意识到路网的复杂性、交通状况的拥挤性、路径选择的随机性和交通需求的动态性对于交通规划、分配、诱导和控制具有重要影响。因此，研究 BRAESS 悖论的机理并避免该现象发生对于交通规划中的路网设计和已有路网上交通流分配策略的制定都大有裨益。

首先从单起点单终点的简单路网入手，分析 BRAESS 悖论的成因

4.2 BRAESS 悖论的出现

交通网络中各路段的容量直接影响到交通流在网络上的分布格局。而 BRAESS 悖论是由于司机从个人利益出发，选择出行成本最小的路径，致使系统达到均衡状态时的总出行成本增加。

1952 年 WARDROP 通过对交通现象的分析提出了关于交通网络流量分布的用户均衡原理[5]：在所有实际使用的路径上，出行成本相等，且小于任何未使

用路径上的出行成本。DAFERMOS 和 SPARROW[5] 首先提出了用户均衡原理与 NASH 均衡原理之间存在关联。在假设交通流量是离散的条件, ROSENTHAL[6] 证明了两者的等价关系。DEVARAJAN[5] 又证明了在连续交通流量的情形下, 满足 WARDROP 用户均衡原理的配流, 等价于一个非合作博弈中 NASH 均衡解。另一方面, 所有合作博弈的解都具有这样的性质: 在均衡点处任何用户收益的增加都必将导致其他用户收益的降低, 网络所能调整的全部合作对策解的集合构成了 PARETO 边界。从博弈论角度来看, BRAESS 悖论实质上是非合作网络中 NASH 平衡点不满足 PARETO 最优性时出现的诡异现象。

4.3 出行时间与流量关系

如图 1 所示的路网, 路段出行时间

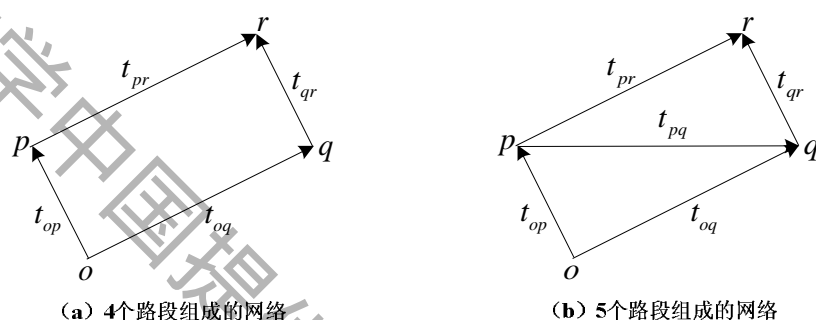


图 1 路网结构图

$$t_{ij} = a_{ij} + b_{ij} f_{ij}$$

其中, t_{ij} 为用户在路段 ij 上的出行时间; a_{ij} 为自由流情况下, 路段 ij 上的出行时间; b_{ij} 为路段 ij 的延迟参数, 即路段 ij 上, 每增加单位流量时, 所增加的出行时间; f_{ij} 为路段 ij 上的车流量。

假定该问题满足对称条件

$$a_{op} = a_{qr}, a_{oq} = a_{pr}, b_{op} = b_{qr}, b_{oq} = b_{pr}$$

进一步假定路网中的瓶颈路段 op 和 qr 很短, 因此自由流情况下的出行时间近似为 0, 即

$$a_{op} = a_{qr} = 0$$

假定新增路段 pq 的延迟参数与路段 oq 和 pr 相等, 即、

$$b_{pq} = b_{oq} = b_{pr}$$

设 $a_{op} = a_{qr} = a_1, a_{pq} = a_2, b_{op} = b_{qr} = b_1, b_{pq} = b_{oq} = b_{pr} = b_2$, 则各段路出行时

间可表示成 $t_{op} = b_1 f_{op}, t_{qr} = b_1 f_{qr}, t_{oq} = a_1 + b_2 f_{oq}, t_{pr} = a_1 + b_2 f_{pr}, t_{pq} = a_2 + b_2 f_{pq}$ 。

路网(A)由 4 个路段构成, 从 O 至 R 有两条路线: 路线 1($o \rightarrow p \rightarrow r$), 路线 2($o \rightarrow q \rightarrow r$), 其出行时间为

$$t^1 = b_1 f_{op} + a_1 + b_2 f_{pr}$$

$$t^2 = b_1 f_{oq} + a_1 + b_2 f_{qr}$$

令 $f_{op} = f_{pr} = x, f_{oq} = f_{qr} = Q - x, x \in [0, Q]$, 并取 $a_1 = 15, a_2 = 7.5, b_1 = 0.01, b_2 = 0$,

$Q = 1000$, 则 $t^1 = 15 + 0.01x, t^2 = 25 - 0.01x, x \in [0, 1000]$ 。作出 $t^1(x), t^2(x)$ 的函数曲线分别如图 2 路网 (A) 中实线和星号线所示。图中两条线的交点为用户均衡解, 偏离这一平衡必然导致一个目标函数值减小时另一个目标函数值增加。此时用户均衡解位于 PARETO 边界上, 因此, 尽管用户之间是非合作关系, 其结果仍满足 PARETO 最优性, 不存在可以同时提高双方性能的解。路网(B) 中除了线路 1、2 以外, 增加了路线 3($o \rightarrow p \rightarrow q \rightarrow r$)。设 f^1 为分配到路线 1 上的交通流, 根据路网对称性, 分配到路线 2 上的交通流也是 f^1 , 因此, 分配到路线 3 上的交通流为 $Q - 2f^1$ 。

取与路网(A)相同的参数并另 $x = f^1$, 做出 $t^1(x)$ 和 $t^3(x)$ 的函数曲线如图 2 路网(B)中实线和星号线所示。

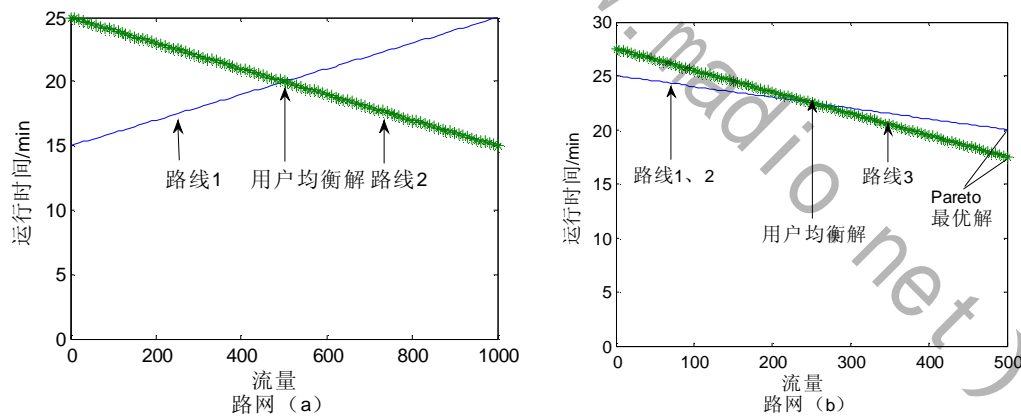


图 2 出行时间与流量关系

此时只有一个 PARETO 最优 $x = 500, t_1 = t_2 = 20$, 路线 3 上没有车流; 而用户均衡解为 $x = 250, t_1 = t_2 = t_3 = 22.5$ 。显然用户均衡解不在 PARETO 边界上, 此时出现 BRAESS 悖论: 在路网(A)中增加一条通路反而增加了所有用户的出行时间。

4.4 北京二环区交通路网中单月的 BRAESS 悖论

实际路网中图 1 所示的结构并不常见, 北京路网中经常见到的结构如图 3 所示。从 O 到 D 有 3 条路线 $O \rightarrow p \rightarrow q \rightarrow D, O \rightarrow r \rightarrow s \rightarrow D, O \rightarrow r \rightarrow q \rightarrow D$ 。设

交通需求为 Q ，分配到 3 条路线上的交通流分别为 x_1, x_2 和 $Q - x_1 - x_2$ 。当

$\frac{b_3}{b_1 + b_2} \geq \frac{b_4 + b_5}{b_6}$ 时，则该问题存在 PARETO 最优解。并且与用户均衡解不一致，

发生 BRAESS 悖论。

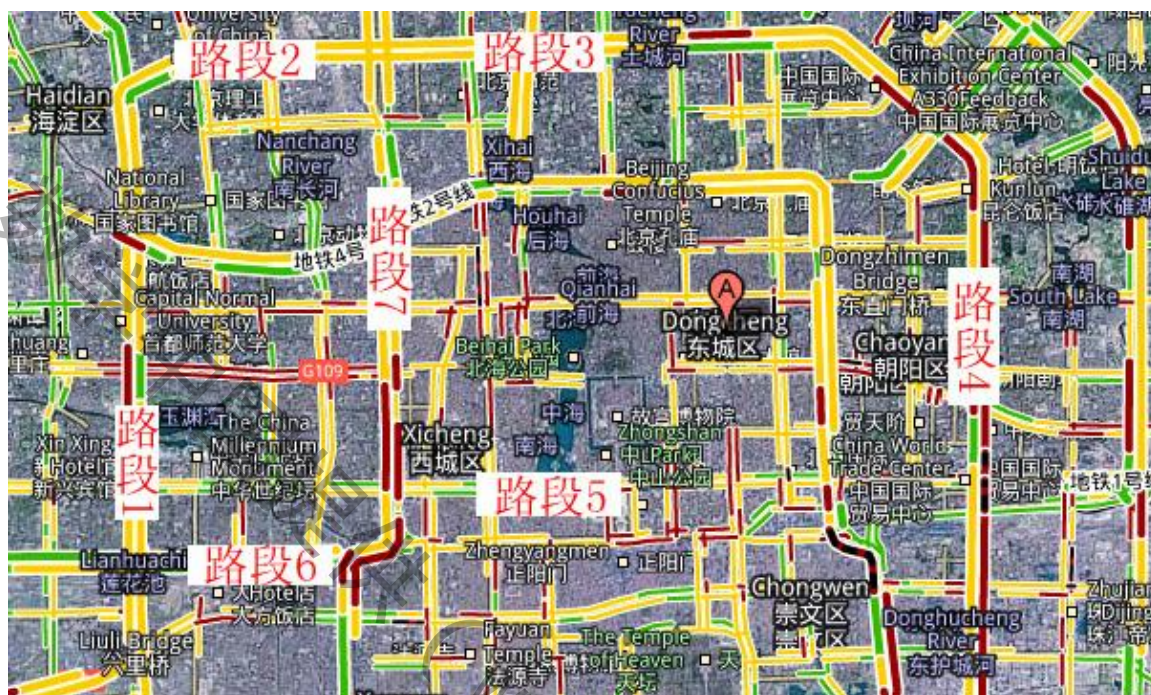


图 3 北京二环区交通路段

分析北京某区域路网。首先，根据路段上一个月的车速 (SPEED) 和流量 (FLOW) 数据以及路段长度 (L)，估计路段的自由流出行时间 a 和延迟参数 b 。

路段出行时间 $t = a + b \cdot flow$ ，出行时间可通过路段长度和车速近似表示

$t = L / speed$ ，因此根据车速和流量的历史数据[7]，可以拟合出 $flow \sim L / speed$ 关

系曲线，从而估计 a 和 b 的值。图 3 中 7 个路段的参数如表 1 所示。

年月	自由流出行时间 (s)	延迟参数
路段 #1	27.19	0.0293
路段 #2	28.16	0.0591
路段#3	19.13	0.1310
路段#4	11.76	0.1192
路段#5	10.27	0.1677
路段#6	26.77	0.0366
路段#7	21.94	0.0293

表 1 自由流出行时间与延迟参数

显然，表 1 给出的路网参数满足

$$\frac{b_3}{b_1 + b_2} \geq \frac{b_4 + b_5}{b_6}$$

这一条件，因此，北京二环交通路网中存在 BRAESS 悖论。

这里，我们以一个月作为衡量范围，对几个路段进行了 BRAESS 悖论分析，但这样的分析具有一定的局限性。若要整体分析北京二环以内是否长期存在这一悖论，我们需要了解路段年延迟参数等情况。

4.5 2000~2009 年北京二环区交通路网中的 BRAESS 悖论

同理，我们以年月为单位，统计关于北京二环以内的交通流量的因素[8]，可以得到相应的综合路段年延迟参数，这里我们计算了 2000~2009 年共 120 个月的延迟参数如下表所示：

年月	自由流出行时间 (s)	延迟参数
2000 #1	22.11	0.0918
2000 #2	29.26	0.1511
2000 #3	28.57	0.0356
2000 #4	31.47	0.1431
2000 #5	15.77	0.0677
2000 #6	36.95	0.1079
2000 #7	11.28	0.0687
2000 #8	18.57	0.0485
2000 #9	23.04	0.0809
2000 #10	22.56	0.0327
2000 #11	36.62	0.0335
2000 #12	22.62	0.1606
2001 #1	30.21	0.0612
2001 #2	27.13	0.1665
2001 #3	23.61	0.1708
2001 #4	19.13	0.0357
2001 #5	25.94	0.0395
2001 #6	20.05	0.0908
2001 #7	11.79	0.0264
2001 #8	10.78	0.1369
2001 #9	13.75	0.0224
2001 #10	32.52	0.1066
2001 #11	31.86	0.0651
2001 #12	31.78	0.0451
2002 #1	29.28	0.1573
2002 #2	11.36	0.1174
2002 #3	23.22	0.0343
2002 #4	28.67	0.1547
2002 #5	23.95	0.0434
2002 #6	34.85	0.1109
2002 #7	20.08	0.0236
2002 #8	32.75	0.0425
2002 #9	39.27	0.1346

报名号#1013

2002 #10	28.24	0.1645
2002 #11	37.05	0.0574
2002 #12	15.08	0.1681
2003 #1	39.79	0.0338
2003 #2	21.86	0.1371
2003 #3	26.78	0.1033
2003 #4	23.88	0.0316
2003 #5	23.86	0.0242
2003 #6	35.04	0.1479
2003 #7	19.83	0.1029
2003 #8	25.19	0.1575
2003 #9	20.37	0.0347
2003 #10	24.91	0.0925
2003 #11	11.96	0.1212
2003 #12	28.04	0.0769
2004 #1	35.46	0.0355
2004 #2	35.33	0.0141
2004 #3	27.65	0.0562
2004 #4	12.47	0.0142
2004 #5	34.81	0.0388
2004 #6	18.93	0.1065
2004 #7	23.84	0.0368
2004 #8	38.68	0.1696
2004 #9	36.17	0.1491
2004 #10	32.92	0.1367
2004 #11	21.81	0.1158
2004 #12	11.14	0.1239
2005 #1	29.91	0.1265
2005 #2	39.22	0.0451
2005 #3	31.8	0.1232
2005 #4	33.64	0.1504
2005 #5	18.39	0.0184
2005 #6	10.11	0.0858
2005 #7	22.29	0.0403
2005 #8	17.71	0.1482
2005 #9	22.24	0.0647
2005 #10	25.47	0.0149
2005 #11	34.42	0.0764
2005 #12	18.95	0.1101
2006 #1	19.48	0.1471
2006 #2	26.14	0.0928
2006 #3	17.35	0.1746
2006 #4	24.82	0.0723
2006 #5	26.02	0.1468
2006 #6	10.31	0.1112
2006 #7	37.23	0.1414

2006 #8	11.65	0.1489
2006 #9	33.34	0.0662
2006 #10	27.84	0.1069
2006 #11	21.44	0.0773
2006 #12	28.23	0.1609
2007 #1	34.66	0.1571
2007 #2	31.02	0.1173
2007 #3	39.33	0.0158
2007 #4	23.21	0.1335
2007 #5	34.47	0.1408
2007 #6	37.91	0.1395
2007 #7	34.09	0.1049
2007 #8	31.19	0.0765
2007 #9	20.29	0.1258
2007 #10	38.59	0.1629
2007 #11	19.74	0.0227
2007 #12	26.97	0.0675
2008 #1	10.94	0.0135
2008 #2	26.63	0.1594
2008 #3	24.94	0.0221
2008 #4	14.68	0.1246
2008 #5	24.17	0.0592
2008 #6	24.76	0.0948
2008 #7	37.95	0.1192
2008 #8	20.44	0.0475
2008 #9	11.94	0.1145
2008 #10	16.43	0.0276
2008 #11	26.07	0.1146
2008 #12	16.05	0.0262
2009 #1	25.82	0.0686
2009 #2	36.61	0.1275
2009 #3	26.99	0.0374
2009 #4	10.47	0.1386
2009 #5	36.58	0.0763
2009 #6	11.32	0.0875
2009 #7	19.88	0.1162
2009 #8	32.35	0.0177
2009 #9	19.95	0.1722
2009 #10	16.44	0.0766
2009 #11	37.48	0.1194
2009 #12	16.26	0.0513

表 2 自由流出行时间与延迟参数

五、NASH 均衡原理与小波神经网络预测模型

5.1 NASH 均衡原理

为考虑北京二环以内的年交通状况是否满足 BRAESS 悖论, 我们引入 DAFERMOS 和 SPARROW[5]提出的 NASH 均衡原理, 引入了路段 NASH 平衡比率的概念, 其含义为: 当某一时间范围内, 路段的路段 NASH 平衡比率低于路段延迟参数时, 其发生了 BRAESS 悖论, 当同时又满足 NASH 平衡波动率超过 30%时, 为严重 BRAESS 悖论, NASH 平衡比率理论方程如下:

$$M(t) = \sum_{t=2000}^T \sum_{k=t+1}^{T-1} \left[\left(\frac{f(t+\Delta t)}{f(t)} - 1 \right) \left(\frac{K(k+\Delta k)}{K(k)} - 1 \right) \right]$$

通过 NASH 平衡比率理论方程及路段延迟系数的数据, 我们可以构建小波神经网络预测模型对路段 NASH 平衡比率进行预测计算。

5. 2 构建小波神经网络对路段 NASH 平衡比率进行预测

连续参数小波神经网络是基于小波分析二构成的具有神经网络思想的模型, 即用连续小波基函数取代了通常的神经网络中的非线性 SIGMOID 函数, 对于路段 NASH 平衡比率组成非线性时间序列, 可以通过用所选取的非线性小波基进行线性叠加来实现, 也就是用小波级数的有限项来逼近路段延迟参数。

5. 2.1 模型的构建

如果小波函数系 $\{j\left(\frac{t-b_k}{a_k}\right); k \in Z\}$ 满足框架条件, 即存在两个常数 $c_{\min} > 0$ 和 $c_{\max} < \infty$, 使得对任意的函数 $f(t) \in L^2(R)$, 下列不等式成立。

$$c_{\min} \|f\|^2 \leq \sum_k |\langle j, f \rangle|^2 \leq c_{\max} \|f\|^2$$

小波函数系 $\{j\left(\frac{t-b_k}{a_k}\right); k \in Z\}$ 在 $L^2(R)$ 空间是稠密的。因此路段 NASH 平衡比率时间序列函数 $f(t)$ 可用连续小波基进行拟合:

$$g(t) = \sum_{k=1}^N W_k j\left(\frac{t-b_k}{a_k}\right)$$

其中 $g(t)$ 为路段 NASH 平衡比率时间序列 $f(t)$ 的预测值序列, W_k 、 a_k 、 b_k 分别为权系数、小波基的伸缩因子、平移因子, N 为小波基的个数, 上述公式的逼近问题可选用连续参数小波神经网络来实现。图 4 给出了连续参数小波神经网络结构, 网络为单隐层结构, 仅含有一个输入和一个输出节点。

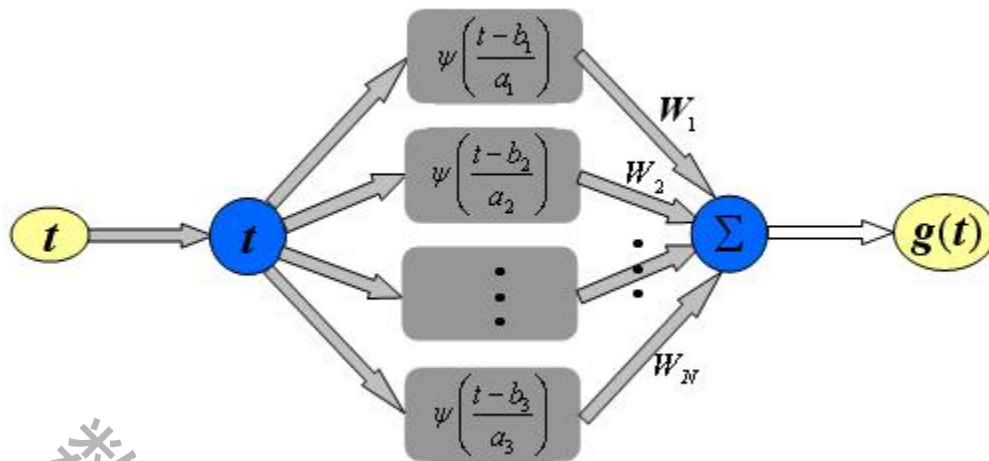


图4 连续参数小波神经网络结构

5.2.2 小波函数的确定

我们的目的是确定网络参数 W_k 、 a_k 、 b_k 可通过最小均方误差能量函数进行优化:

$$E = \frac{1}{2} \sum_{t=1}^N [f(t) - g(t)]^2$$

其中 N 为数据采样总数, 小波函数可采用国外较多使用的 MORLET 母小波, 该小波是余弦调制的高斯波 $f(t) = \cos(1.75t) \exp(-t^2/2)$ 。为了书写方便,

令 $t' = \frac{t-b_k}{a_k}$, 将小波函数带入上述能量函数, 则 E 的梯度可分别表示为:

$$g(W_k) = \frac{\partial E}{\partial W_k} = - \sum_{t=1}^N [f(t) - g(t)] \cos(1.75t') \exp\left(-\frac{t'^2}{2}\right)$$

$$g(b_k) = \frac{\partial E}{\partial b_k} = - \sum_{t=1}^N [f(t) - g(t)] W_k \left[1.75 \sin(1.75t') \exp\left(-\frac{t'^2}{2}\right) \frac{1}{a_k} + \cos(1.75t') \exp\left(-\frac{t'^2}{2}\right) \frac{t'}{a_k} \right]$$

$$g(a_k) = \frac{\partial E}{\partial a_k} = - \sum_{t=1}^N [f(t) - g(t)] W_k \left[1.75 \sin(1.75t') \frac{t'}{a_k} + \cos(1.75t') \exp\left(-\frac{t'^2}{2}\right) \frac{t'^2}{a_k} \right] = t' g(b_k)$$

采用共轭梯度法优化网络参数 W_k 、 a_k 、 b_k , 分别令向量: $\dot{W} = (W_1, W_2, \mathbf{L}, \mathbf{L}, W_k)$, $\dot{g}(\dot{W}) = (g(W_1), g(W_2), \mathbf{L}, \mathbf{L}, g(W_k))$, $\dot{S}(\dot{W})_i$ 为 \dot{W} 第 i 次循环搜索方向, 它是 W 的函数, 则:

$$\dot{S}(\dot{W}) = \begin{cases} -\dot{g}(\dot{W})_i \\ -\dot{g}(\dot{W})_i + \frac{\dot{g}(\dot{W})_i [-\dot{g}(\dot{W})_i]^T}{\dot{g}(\dot{W})_{i-1} [-\dot{g}(\dot{W})_{i-1}]^T} \dot{S}(\dot{W})_{i-1} \end{cases}$$

权重向量按下式调节:

$$\dot{W}_{i+1} = \dot{W}_i + a_w \dot{S}(\dot{W})_i$$

第一步是计算第 i 此迭代的以搜索方向 S ; 第二步则用变步长 a_w 计算的权值矢量, 对每一步迭代, 计算步长的选择由两种方式: 变步长和固定步长。

在此采用一维搜索变步长法计算最佳步长 a_w , 每次循环时, 按上述两个公式分别调节网络参数向量 W , 直至收敛与某一确定值或循环结束为止, 对参数 a_k 、 b_k 的训练, 只要把上述两个公式中的 W 换为 a_k 、 b_k 之后, 即可按同样方

式调整。

5.2.3 对隐层小波元的确定

本文中我们将选择 2000~2009 年路段 延迟参数为样本，根据上述 NASH 平衡原理，路段延迟参数与路段 NASH 平衡比率的动态预测关系式，以小波神经网络的能源方程为基础对 2000~2009 年路段 NASH 平衡比率进行计算，模型中我们将隐层小波元选为 10 个。

5.2.4 模型的求解

根据上述模型提供的算法，利用 MATLAB 编程得到权值矩阵 W 如下：

W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_8	W_9	W_{10}
32122	37468	78155	50168	89526	37598	70654	31468	80955	72945

其中 $a_k=0.23532$, $b_k=0.623782$

得到 W_k 、 a_k 、 b_k 的值后，我们可以计算出 2000~2009 年各路段 NASH 平衡比率，这里我们提取出 2002 年（6 月天坛路拓宽），2003 年（6 月普方路拓宽），2004 年（7 月永定门桥延修），2008 年（9 月天坛路改道）的 NASH 平衡比率，并通过 MATLAB 工具 CHIP（）提取出月波动率超过 30% 的比率数进行标定（蓝色线），同时，我们将近十年的北京二环以内主要路段的 BRAESS 悖论情况作如下标定：（红星代表一年中出现 BRAESS 悖论超过六个月的路段，蓝星反之）。



图 5 2000 年北京二环路段 BRAESS 悖论情况

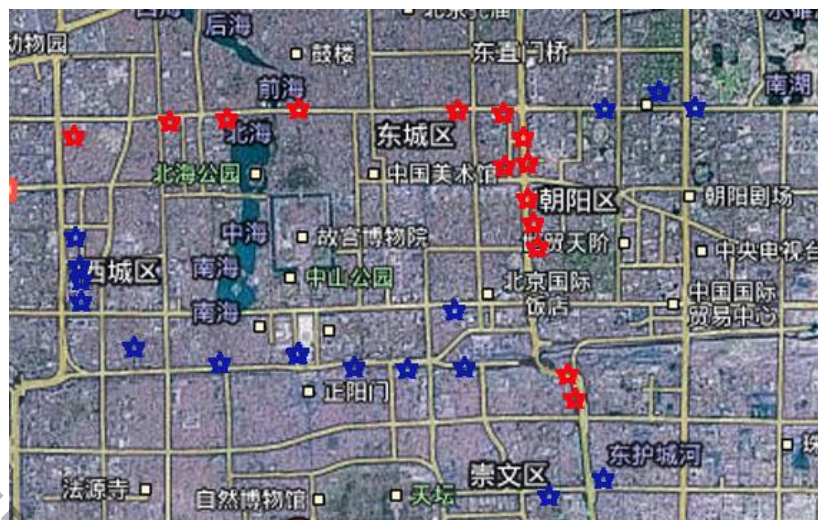


图 6 2001 年北京二环路段 BRAESS 悖论情况



图 7 2002 年北京二环路段 BRAESS 悖论情况



图 8 2003 年北京二环路段 BRAESS 悖论情况



图 9 2004 年北京二环路段 BRAESS 悖论情况



图 10 2005 年北京二环路段 BRAESS 悖论情况



图 11 2006 年北京二环路段 BRAESS 悖论情况



图 12 2007 年北京二环路段 BRAESS 悖论情况

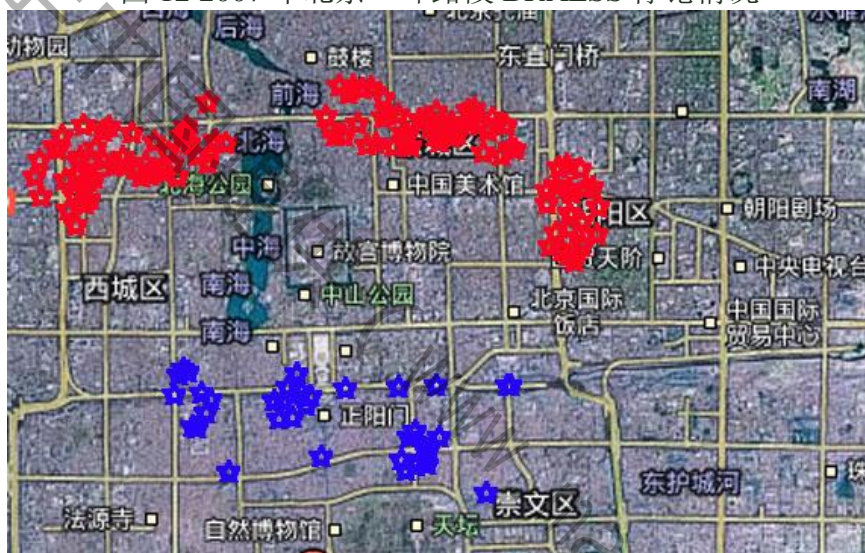


图 13 2008 年北京二环路段 BRAESS 悖论情况



图 14 2009 年北京二环路段 BRAESS 悖论情况

5.3 结论一：近十年北京二环路段 BRAESS 悖论情况

东城区，朝阳区路段常年持续存在 BRAESS 悖论情况（平均每年均超过 6 个月）。事实上，尽管我们发现 2002 年（6 月天坛路拓宽），2003 年（6 月普方路拓宽），2004 年（7 月永定门桥延修），2008 年（9 月天坛路改道）会给崇文区与西城区南部路段带来一定的交通便利，但结果显示，其整个北京二环路段的交通情况的平衡性变得更不稳定，交通网络中 NASH 平衡点不满足 PARETO 最优性的问题更加突出。譬如在 2006 年，2009 年的东城区，朝阳区路段堵塞问题更加尖锐，这也使得本来交通相对通畅的南段也出现 BRAESS 悖论情况。

5.4 结论二：NASH 平衡比率与路段延迟系数的模拟结果

我们提取出 2002 年（6 月天坛路拓宽），2003 年（6 月普方路拓宽），2004 年（7 月永定门桥延修），2008 年（9 月天坛路改道）的 NASH 平衡比率（绿线），相应的路段延迟系数（红线），并通过 MATLAB 工具 CHIP（）提取出月波动率超过 30% 的比率数进行标定（蓝色线）如下：

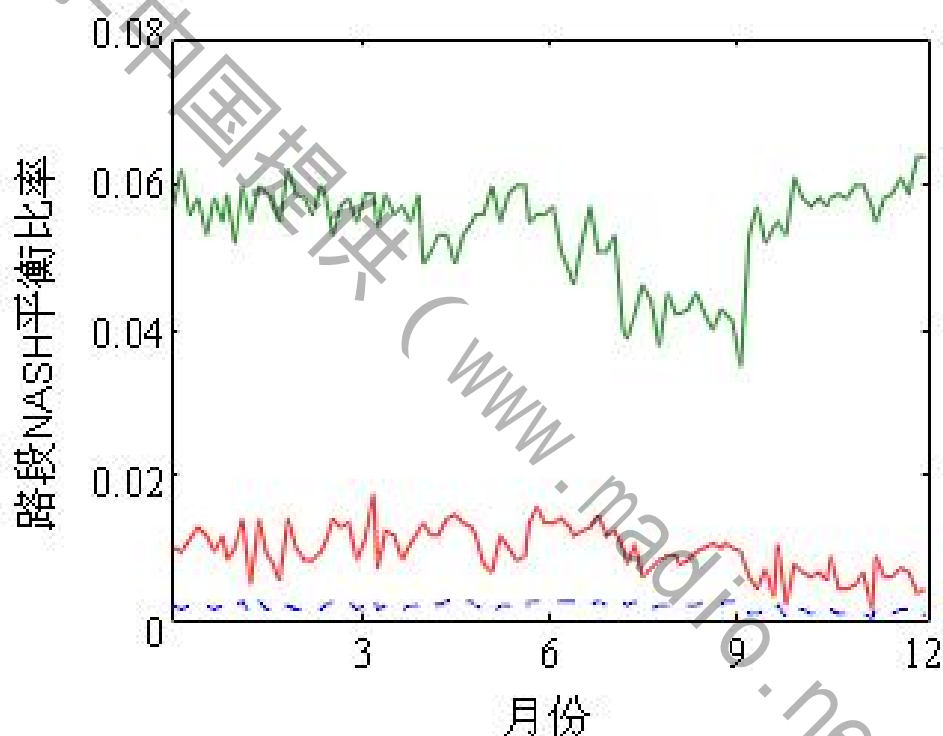


图 15 2002 年北京二环路段 NASH 平衡比率

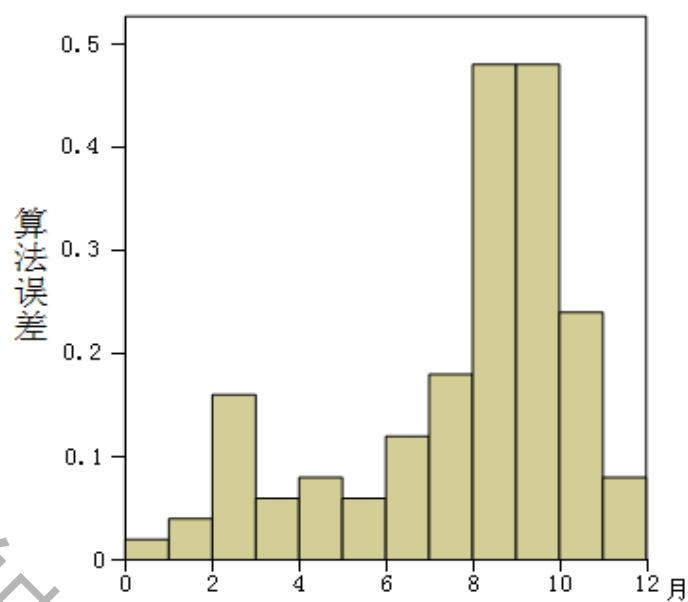
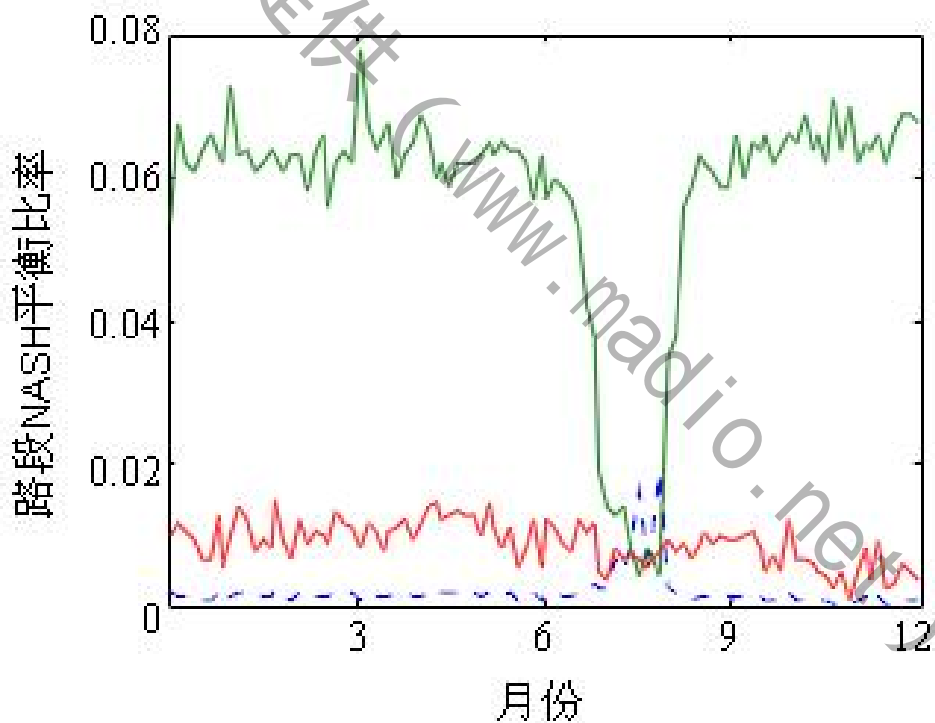


图 16 算法误差

2002 年总体来看，北京二环路段并未出现严重的 **BRAESS** 悖论情况，尽管在 7~9 月，路段 **NASH** 平衡比率有所下降，但并未低于延迟系数，波动情况稳定。

图 17 2003 年北京二环路段 **NASH** 平衡比率

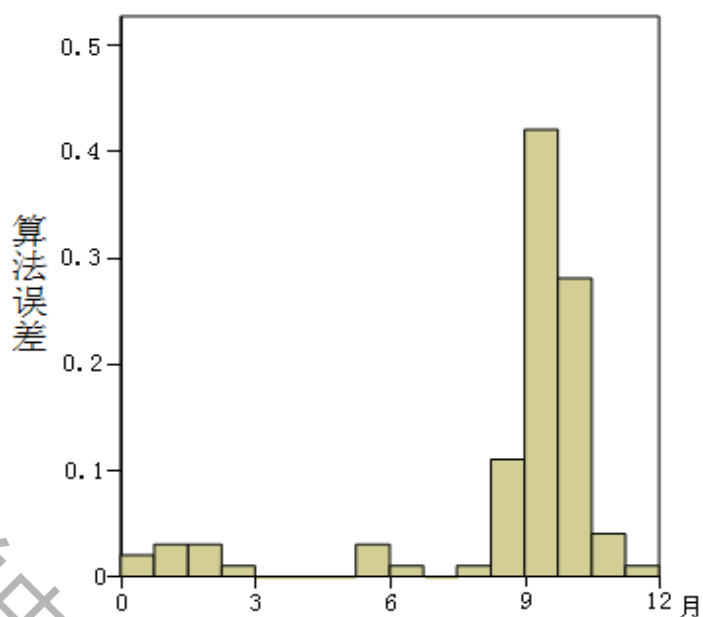


图 18 算法误差

2003 年总体来看，北京二环路 7、8 月份出现严重的 BRAESS 悖论情况，路段 NASH 平衡比率大幅下降，低于延迟系数，波动情况约为 34%，属于严重的 BRAESS 悖论情况。

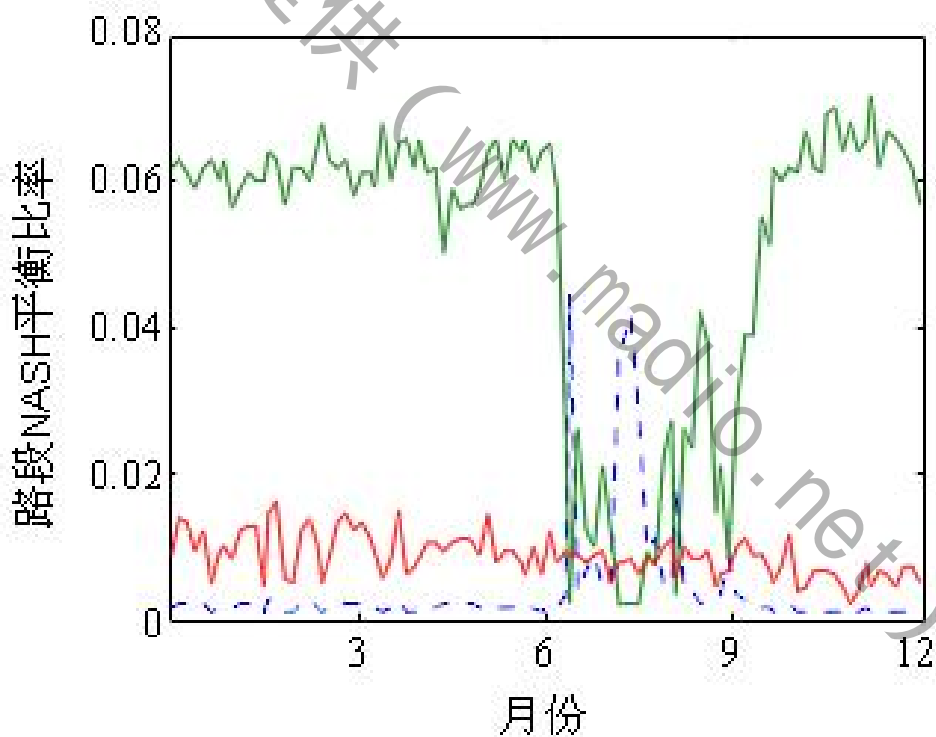


图 19 2004 年北京二环路 路段 NASH 平衡比率

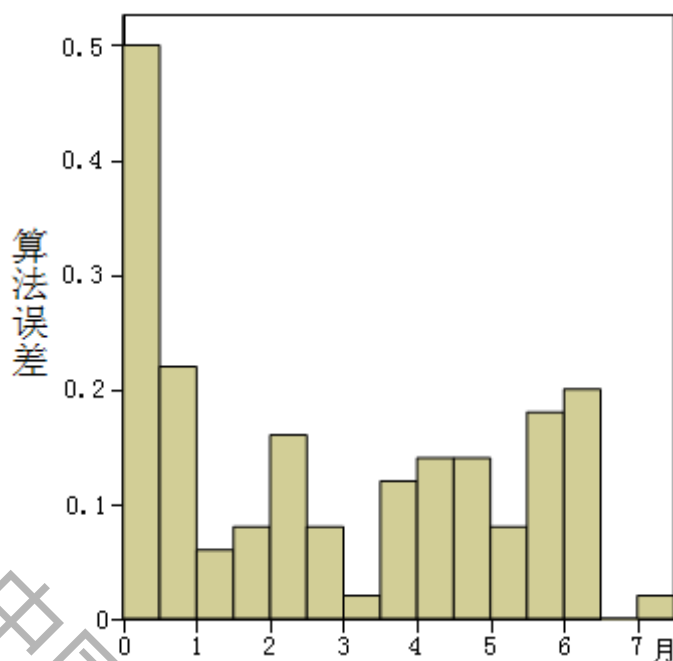


图 20 算法误差

2004 年总体来看，北京二环路 6, 7, 8, 9 月份均出现严重的 BRAESS 悖论情况，全年路况糟糕，路段 NASH 平衡比率最低达 0.0014，低于延迟系数，波动情况约为 56%，属于严重的 BRAESS 悖论情况。

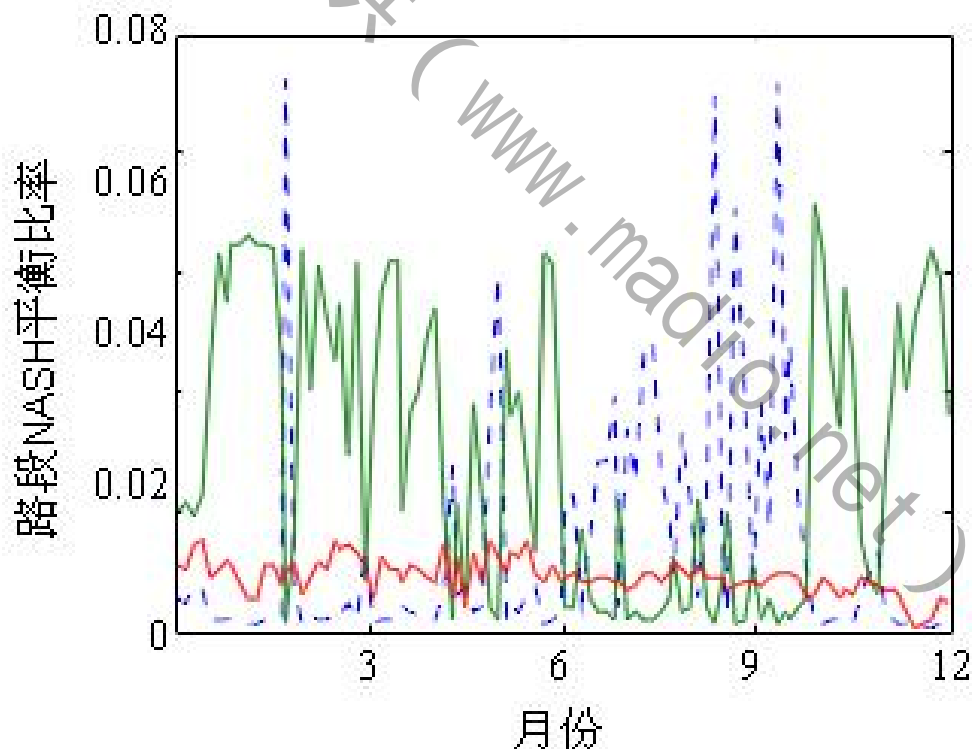


图 21 2008 年北京二环路 NASH 平衡比率

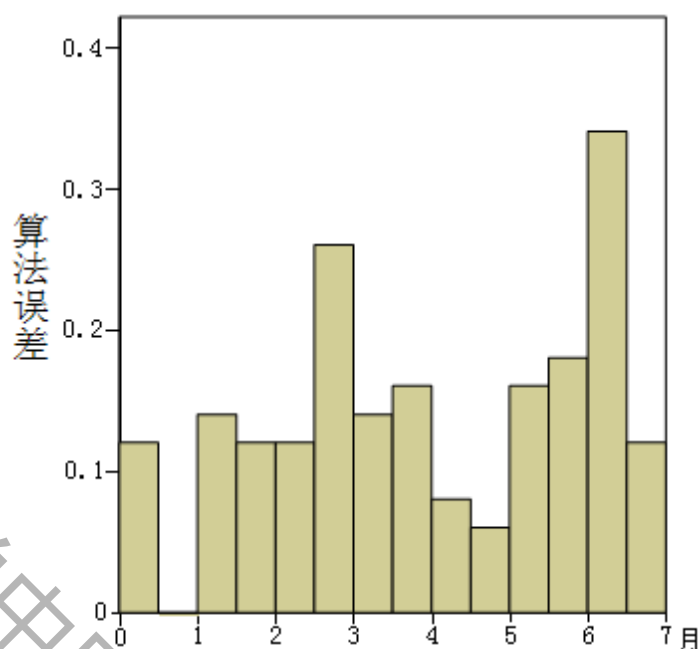


图 22 算法误差

2008 年总体来看,北京二环路只有 1, 12 月份没有 BRAESS 悖论情况,路段 NASH 平衡比率多次大幅低于延迟系数,波动情况最大达 81%,属于严重的 BRAESS 悖论情况。

六、基于 GPS 动态导航的车辆运行模型

6.1 GPS 交通系统的动态随机特性

GPS 交通系统是一个由人、GPS 系统、车辆和公路所构成的复杂大系统,其复杂性也就决定了车辆导航的复杂性。为了对车辆进行合理的导航,首先需要对 GPS 交通系统进行分析,以便考察其动态随机的特性。

6.2 车辆导航用户

用户自身的一些特性会影响到行车状况,如用户车辆的特性和用户的心理状态都会影响到车辆的运行速度。依据对未来交通状况的预测,为用户提供可供参考的导行车速,将有助于提高用户行车的安全性,并能有效降低用户行车的随意性。

通过以上分析可以看到,尽管交通系统极其庞大,富于变化,但其变化大致能被预测到。其中,不能被预测到的小波动,可以看作随机扰动。另外,交通系统还具有一定的惯性,难以在极短的时间内发生变化。

交通系统的特性决定了对车辆的导航必然是在动态随机状态下的导航。在此状态下的导航既要把握住交通状况的变化趋势,对未来的交通状况进行可靠的预测,又要详细地研究交通的随机特性,正确地刻画随机因素的分布特征。

6.3 动态随机特性的数学描述

由以上对 GPS 交通系统的分析可知,可以用普通变量和随机变量来综合描述各种因素对用户行车的影响。依据车辆运行的特性,便可以建立起车辆运行速度的数学模型。将实时测得的各种参数带入模型,并结合用户自身特性[9],可以估计其在未来某时刻路网的各路段的导行车速,再结合各路段的长度,便可以准确地预测用户通过各路段的时耗。

以 t_{ij} 表示用户通过路段 (i, j) 的时耗(即行驶时间)。由上述分析可知它是一个随机变量。根据对交通系统的分析可以认为,系统中发生的较大波动都能被预测到,而无法预测的都是小波动,它们通常只影响本路段内的车辆。因此,对于路网中的两条路段 (i, j) 和 (l, m) , 其时耗的概率密度函数满足下述关系:

$$f_{t_{ij}, t_{lm}}(x, y) = f_{t_{ij}}(x)f_{t_{lm}}(y)$$

其中 $f_{t_{ij}, t_{lm}}(x, y)$ 是 t_{ij}, t_{lm} 的联合概率密度函数。

根据数学期望的定义有:

$$\begin{aligned} E(t_{ij}, t_{lm}) &= \iint_D xyf_{t_{ij}, t_{lm}}(x, y)dxdy = \iint_D xyf_{t_{ij}}(x)f_{t_{lm}}(y)dxdy \\ &= \int_{-\infty}^{+\infty} xf_{t_{ij}}(x)dx \int_{-\infty}^{+\infty} yf_{t_{lm}}(y)dy = E(t_{ij})E(t_{lm}) \end{aligned}$$

于是, 对于路段时耗的协方差有:

$$\text{Cov}(t_{ij}, t_{lm}) = E\{[t_{ij} - E(t_{ij})][t_{lm} - E(t_{lm})]\} = 0$$

根据方差的定义有:

$$\begin{aligned} D\left(\sum_{(i,j) \in P} t_{ij}\right) &= E\left[\sum_{(i,j) \in P} t_{ij} - E\left(\sum_{(i,j) \in P} t_{ij}\right)\right]^2 \\ &= \sum_{(i,j) \in P} E[t_{ij} - E(t_{ij})]^2 + 2 \sum_{\substack{(i,j) \in P \\ (l,m) \in P}} \text{Cov}(t_{ij}, t_{lm}) \\ &= \sum_{(i,j) \in P} E[t_{ij} - E(t_{ij})]^2 = \sum_{(i,j) \in P} D(t_{ij}) \end{aligned}$$

即有下式成立:

$$D\left(\sum_{(i,j) \in P} t_{ij}\right) = \sum_{(i,j) \in P} D(t_{ij})$$

其中 P 是路网中任意一条路径。

由于路段的时耗方差在短期内不会变化, 故在路径寻优时可将其看作常量。上面的式子可改写为:

$$D_P = \sum_{(i,j) \in P} D_{ij}$$

式中:

$$D_P = D\left(\sum_{(i,j) \in P} t_{ij}\right), D_{ij} = D(t_{ij})$$

由数学期望的性质可知下式成立:

$$E\left(\sum_{(i,j) \in P} t_{ij}\right) = \sum_{(i,j) \in P} E(t_{ij})$$

或

$$T_P = \sum_{(i,j) \in P} T_{ij}$$

式中：

$$T_P = E\left(\sum_{(i,j) \in P} t_{ij}\right), T_{ij} = E(t_{ij})$$

对于路网中的任意节点 i 及其后继节点 j ， P_i 和 P_j 分别是自起始节点 S 到节点 I 和节点 J 的某一路径且满足条件 $P_i \subset P_j$ 。 T_{P_i} 和 T_{P_j} 分别表示自起始节点 S 沿路径 P_i 和 P_j 到达节点 I 的时耗期望值和方差，且令 $T_S = 0, D_S = 0$ ，根据 D_P 和 T_P 的表达式有：

$$T_{P_j} = \sum_{(l,m) \in P_j} T_{lm} = \sum_{(l,m) \in P_i} T_{lm} + T_{ij} = T_{P_i} + T_{ij}$$

$$D_{P_j} = \sum_{(l,m) \in P_j} D_{lm} = \sum_{(l,m) \in P_i} D_{lm} + D_{ij} = D_{P_i} + D_{ij}$$

总结上述推理，有以下等式成立：

$$T_{P_j} = T_{P_i} + T_{ij}$$

$$D_{P_j} = D_{P_i} + D_{ij}$$

道路网的用户总是希望尽可能地减少路径的时耗期望值和时耗方差。然而，在现实中，这两个目标常常相互矛盾，而呈现出此增彼减的情形。一个比较现实的思路是寻求二者之间的相对平衡，比如，可在一定的时耗方差约束下寻求具有最小时耗期望值的路径。

6.4 动态随机状态下的路径寻优

当时耗在路网中各路段的随机波动不大时，即当各路段的时耗方差很小时，可暂时忽略方差，而仅考虑用户在各路段的时耗期望值。此时的路径寻优，就是寻求具有最小时耗期望值的路径。由于各路段的时耗期望值 $T_{ij}(t)$ 随时间 T 而变化，故而用于静态寻优的 DIJKSTRA 算法不再适用。对其进行适当的改造，可获得较为理想的算法。

6.5 时耗期望值最小的路径求解算法

用 OPEN 表存放已经产生但还没有扩展的节点，用 CLOSED 表存放已经扩展的节点。求解时耗期望值最小路径的算法可表述如下。

- 1) 设初始表中仅含起始点 S ，且 $T_S = 0$ ，对于其余节点有 $T_i = \infty$ ；
- 2) OPEN 表为空，则出错；否则，从 OPEN 表中选择具有最小时耗期望值的节点 I ，设其为 BEST。确认 BEST 是否是目标节点，如果是，转步骤 4)；否则，根据其在路网中的连接路段属性生成节点 I 的后继节点 J 。对于每一后继节点 J ，如果它不和 CLOSED 表中的节点相匹配，则完成下列步骤：
 - ①由 T_{P_j} 的表达式计算节点 J 的时耗期望值；
 - ②如果节点 J 已和 OPEN 表中的一个节点相匹配，检查节点 J 是否具有较小的时

耗期望值,如果 J 节点的值较小,则用节点 J 的值代替匹配节点的值,然后设置匹配节点的后向指针指向BEST节点;

③如果节点 J 不在OPEN表中,设置节点 J 的后向指针指向BEST节点,然后将节点 J 放入OPEN表中;

3) 从OPEN表中移出节点BEST,加入CLOSED表中,重复步骤2);

4) 从BEST节点,遍历后向指针到原节点,报告路径解。BEST节点的值就是用户从起始点到终点的最小耗时期望值。

5) 依据求出的解路径,计算路径解的耗时方差 D_p 。由标准正态分布计算其置信

度为 $1-a$ 的单侧估计区间 $(0, T_p + Na\sqrt{D_p})$, 其中 $\Phi(Na)=1-a$ 。该置信区间表

明若用户选择解路径,有 $1-a$ 的概率保证用户能在 $T_p + Na\sqrt{D_p}$ 时间以内到达终点。

算法中的步骤2)不断地优化着从起始节点到其余节点的路径,保证了算法的正确性。此外,算法不仅求出了具有最小耗时期望值的路径解,还计算出刻画路径耗时统计特征的单侧置信区间,以便于用户更好地进行抉择。

有了车辆运行模型的最小的路径求解算法,我们就可以进一步利用这些目标函数来作为ATM路由选择的约束条件,从而完成了ATM路由选择的影响因子限定。

七、基于ATM路由选择对GPS动态导航对路段影响的预测

7.1 ATM路由选择

作为一种高速通信网,ATM网络以面向连接的方式向用户提供服务因此为ATM呼叫连接请求进行合理的路由选择就成为ATM网络中的关键技术之一,而由此路由选择所进行的最优化判断分析已成为在科研中最为有效的方法之一。

为提高ATM网络资源利用率,综合多种统计复接的方式,使得链路可用资源以及交换节点可用资源动态改变,造成路由选择的困难,这里我们将网络中的路段延迟系数,路段NASH平衡比率, GPS动态导航使用率等性能指标为GPS导航系数,使ATM路由选择问题可以借鉴一些传统路由选择方法,这里我们提出ATM信源等效路径。其概念等效路径的意义是,当业务源输入各路段时速,流量, GPS动态导航使用率等信息时,与其GPS导航系数相对应的路径选择速率等性能又与输入路段NASH平衡比率,路段延迟系数密切相关。这样只要网络业务源能保证提供给精确的路段NASH平衡比率, GPS动态导航使用率就能相应地保证路由选择的最优化。通过引入等效路径的概念,可以利用一般网中路由选择的方法实现ATM网络中路由选择。

当前,大部分实际运行的网络采用最优路径算法或其某种变型最优路径选择要求通过多转接选择使路径总体复杂度达到最小传统的最优路径选择方法可分为无路由表法及有路由表法这两种方法均可运用到ATM路由选择中。

这里我们采用有路由表法时,当信息到达中间节点后,沿着该节点路由表所提供的信息来选择复杂度最小的一条输出链路,以到达下一个节点,再以次循环即得到最优路径。

尽管路由表法在实际中得到了广泛的应用，但它有如下不足，路由表根据一定规则更新，有时无法跟上网络状态的变化，不能对网络变化作出及时的反应。当网络规模增大时，源-目的节点的路径数急剧增大，为保证路由选择的有效性，交换节点中保存的路由表将随之迅速增大，这对路由表的保存和更新带来较大困难。利用人工神经网络大规模并行处理的特点可以有效地解决这一问题。

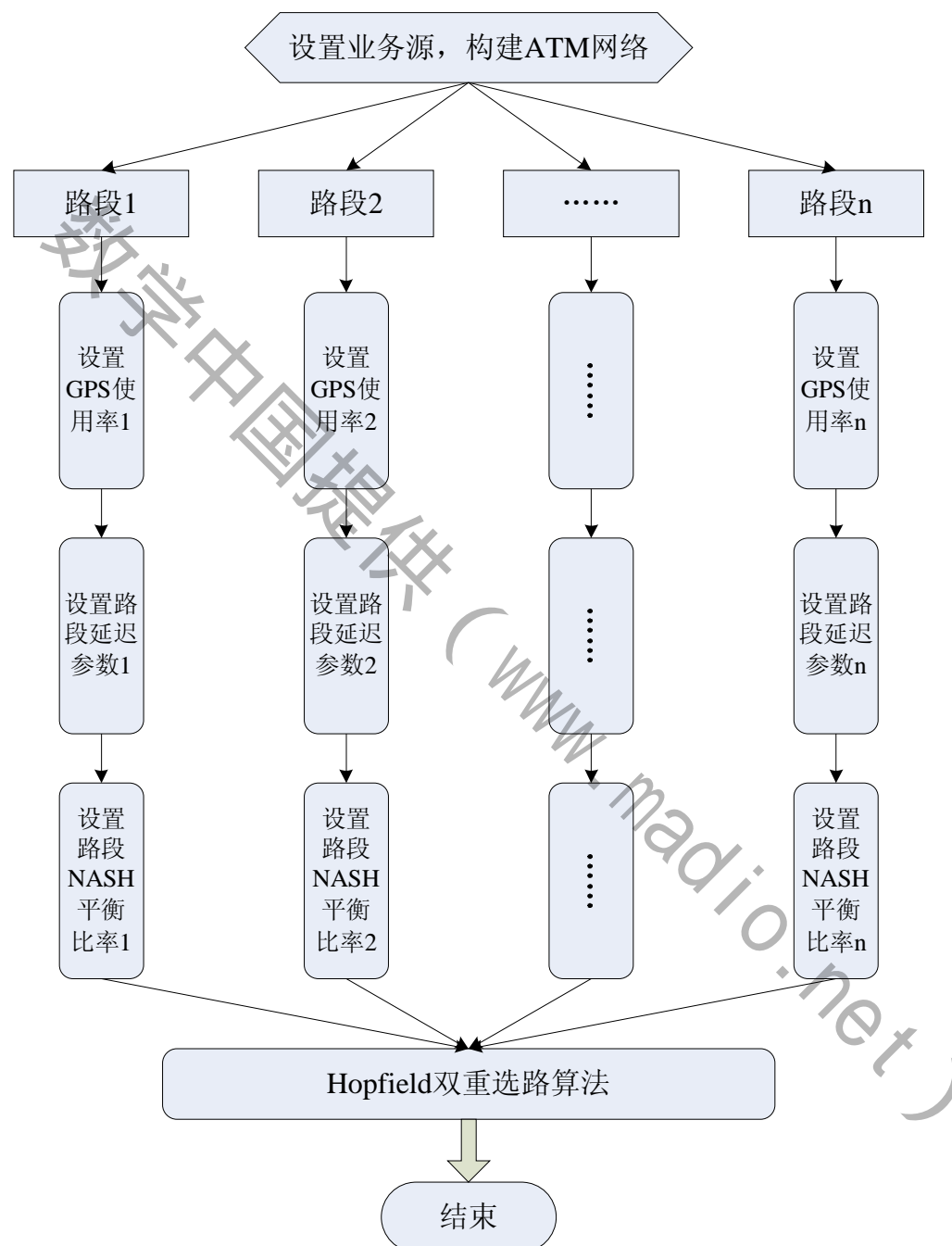


图23 模拟的思想流程

7.2 用人工神经网络方法求解ATM路由选择问题

当给定业务源时，路由选择就是找出一条源-目的间的通路，该通路应能满足连接请求的性能要求。这样的通路对一个业务源请求可很多，但选择不同路径对GPS导航系数有不同影响。因此，一般应采用某种优化路径选择优化路径选择算

法的实现，要依据网络和网络状态的描述参数以及优化策略和优化目标实现路由选择。在本模型中，我们以提升路段 NASH平衡比率为目标。使业务量尽可能的分布在网络中。连路上的流量为各路径有效权值之和。上述框图为本模型模拟的思想流程。

这里我们给出如下的约束条件：

- (1) 当路段GPS导航使用率高于0.34，在模拟系统中相应的路径权值要提升10%，当GPS导航使用率低于8%，路段延迟参数超过0.15时，在模拟系统中相应的路径权值要降低10%，以体现动态导航对于路段状况的优化选择。
- (2) 以第六段GPS动态导航的车辆运行模型的目标函数为路由选择的最优值，以该节点所对应多条路径中的最小值为优化方向
- (3) 路径中的业务源设置北京二环以内的路段信息，逐步分析预测若提高GPS导航使用率对交通环境的影响情况。
- (4) 网络的最终目标是提高路段 NASH平衡比率

为保证HOPFIELD网络能有效完成优化计算，定义如下的能量函数

$$E = \frac{A}{2} \sum_x (\sum_i V_{xi} - 1)^2 + \frac{B}{2} \sum_x \sum_i \sum_{j \neq i} V_{xi} V_{xj} + \frac{C}{2} \sum_r \sum_x \sum_{y \neq x} V_{xr} V_{yr} + \frac{D}{2} \sum_r \sum_{y \neq z} \sum_i F(x, y) V_{ri} (V_{y, i+1} + V_{y, i-1})$$

由此能量函数构造的HOPFIELD网络，连接权重和流量为

$$T_{xi, yj} = -Ad_{xy} - Bd_{xy}(1 - d_{xy}) - Cd_{ij}(1 - d_{xy}) - DF(x, y)(d_{j, i+1} + d_{j, i-1})$$

$$I_r = A$$

其中： $F(x, y) = F_0 + r^2$, $r = a_{xy} / c_{xy}$, $V_{xi} = 1$ 表示节点X为路径中第I个转接点

神经元特性函数选择分段线性函数

$$V(x) = \begin{cases} 0 & x < -k \\ 0.5 + kx & -k \leq x < k \\ 1 & x > k \end{cases}$$

能量函数结构的前三项为约束项，最后一项为优化项。实际工作中，可根据路径长度选取不同的A, B, C, D参数值。当路径跳数较大时，约束项参数值A, B, C相应增大，优化项参数值D相应减小

7.3 HOPFIELD网络的双重选路法

从计算机模拟中可知，HOPFIELD网络能够有效解决网络的路由选择问题，网络本身的并行处理能力是有效解决问题的保证。网络可以实际解决一些实时性要求不高的问题为能利用网络求解优化问题的优势，同时不使运算量增加过多。我们提出双重选路法，双重选路法的基本思路是根据不同的网络状态，采用不同的选路方法，达到兼顾算法复杂度和所选路径最优性之间的折衷具体来说，即依据以下几个步骤进行路由选择，（1）利用传统方法对网络中每对源-目的节点对求解出最短转接次数路径，构造一张静态路（2）由表当业务源到来时，在路由表中查出源一目的节点间的路径，分别计算构成路径的各链路上负载（3）当每条链路负载低于一定程度时，认为网络处于轻载状态，仅选用静态路由表中的最短路径（4）当构成最短路径的链路中有至少一条链路负载超过某一闭值时，认为网络处于重载状态。

此时启动双重最优选路方法。
双重选择排序算法框图如下：

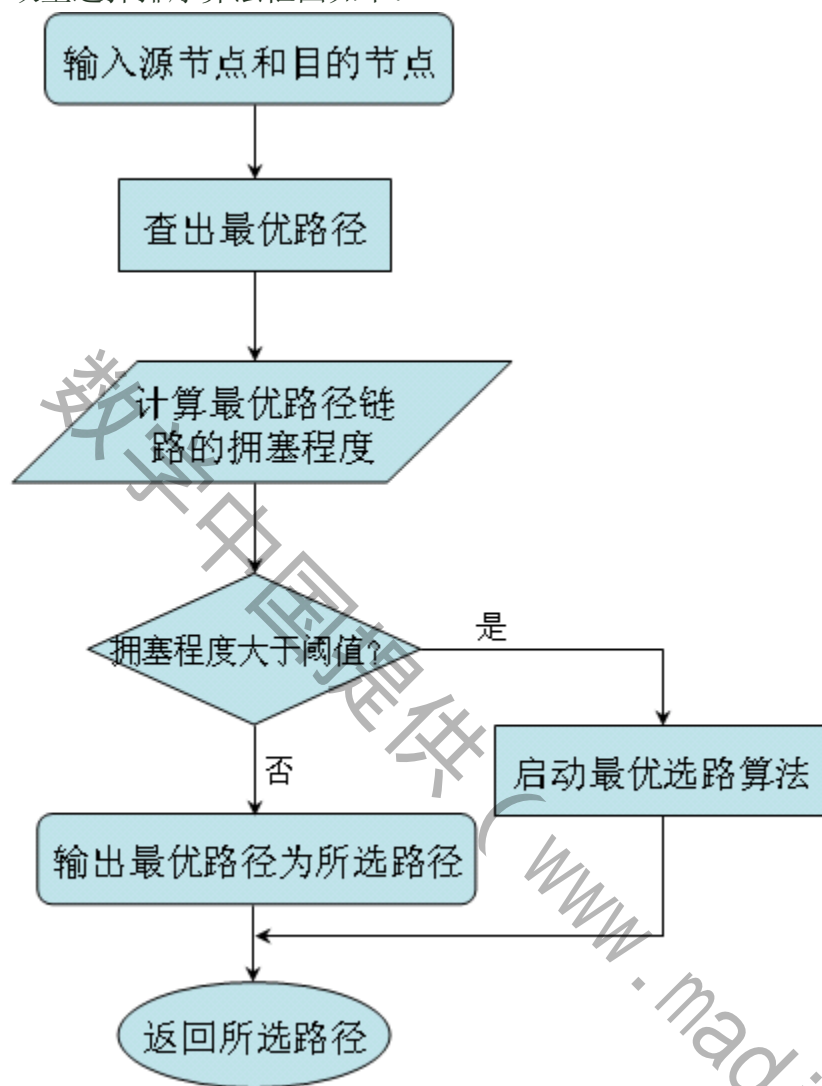


图24 双重选择排序算法框图

7.4 算法模拟预测结果：

得到的在GPS影响下，2002，2003，2004，2008年路段NASH平衡比率及耗时方差的预测结果如下：

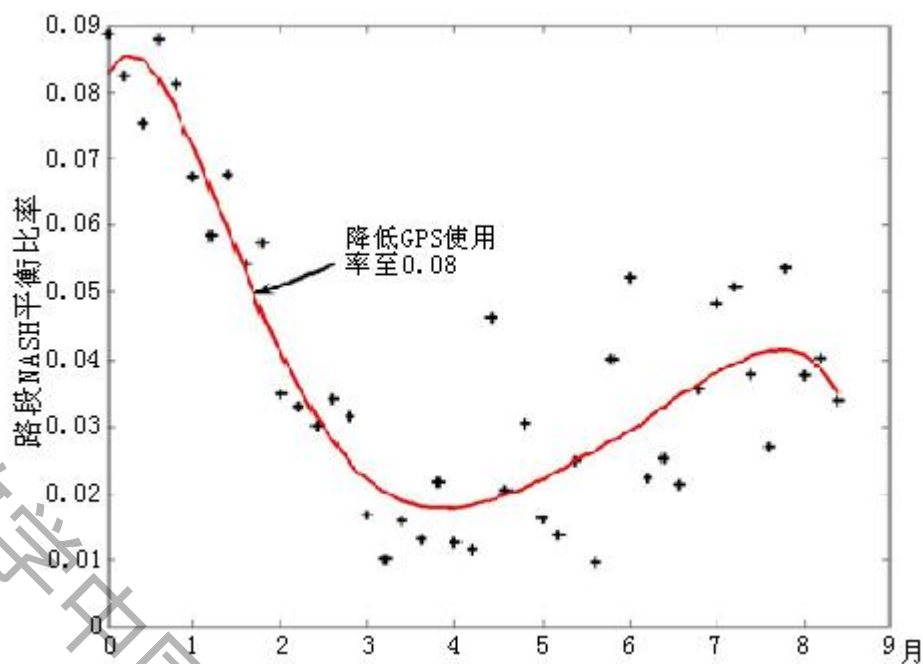


图25 2002年GPS影响下路段NASH平衡比率

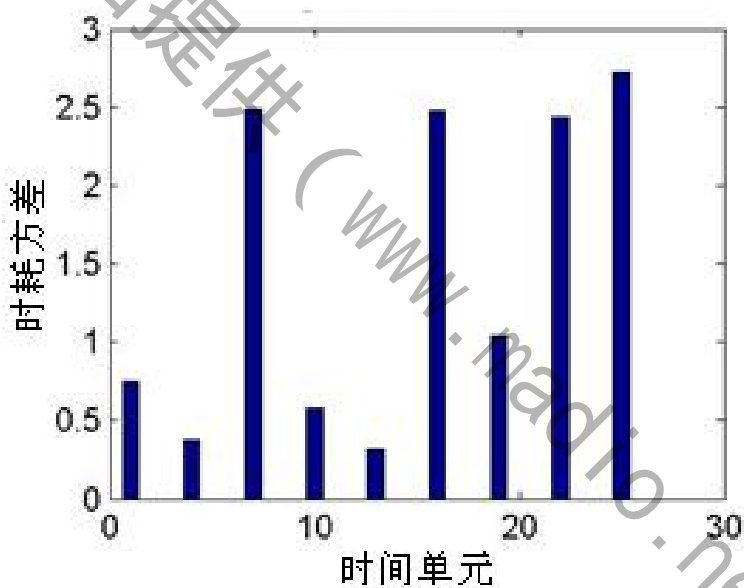


图26

在2002年，当我们持续控制业务源降低GPS使用率时，我们发现路段NASH平衡比率也持续下降，尤其当使用率降至0.08以下时，NASH平衡比率下降幅度最大化。同时算法也兼顾了对时耗方差的控制。

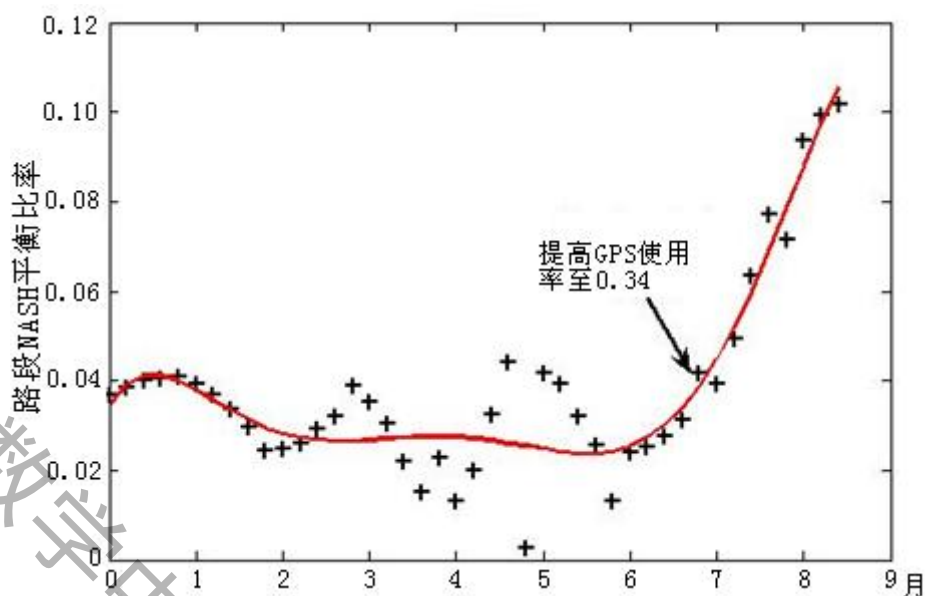


图27 2003年GPS影响下路段NASH平衡比率

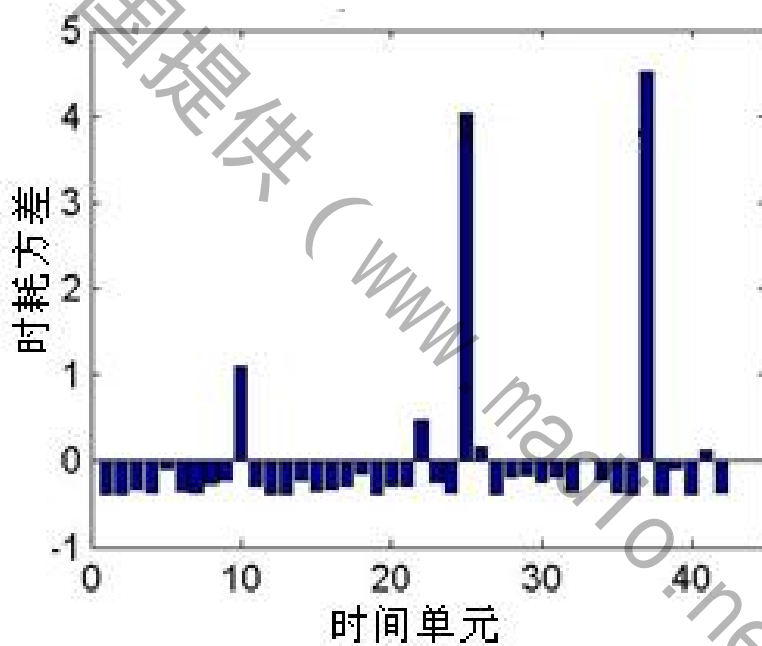


图28

在2003年6月，对应普方路拓宽，当我们持续控制业务源提高GPS使用率时，我们发现路段NASH平衡比率也持续上升，尤其当使用率升至0.34时，NASH平衡比率上升幅度最大化。同时算法也兼顾了对时耗方差的控制。

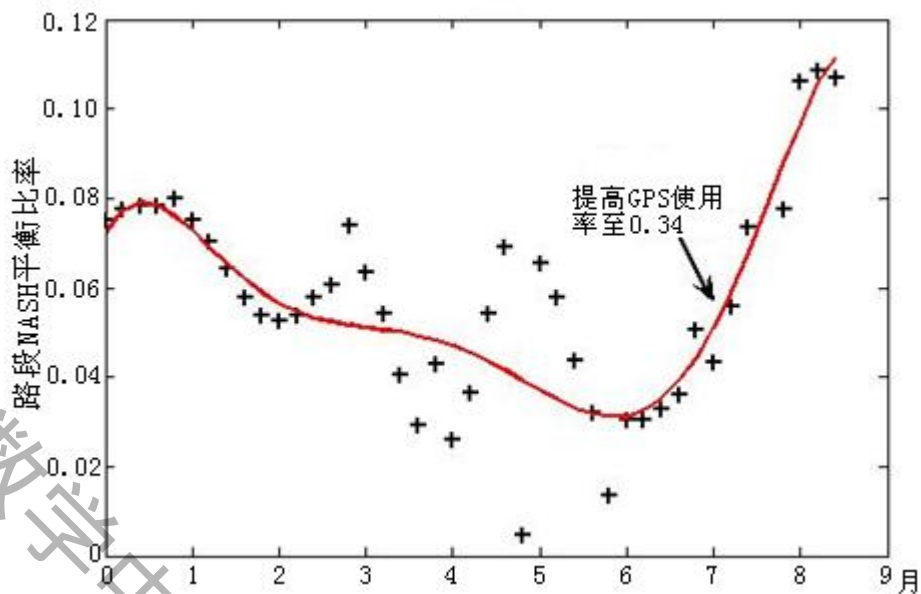


图29 2004年GPS影响下路段NASH平衡比率

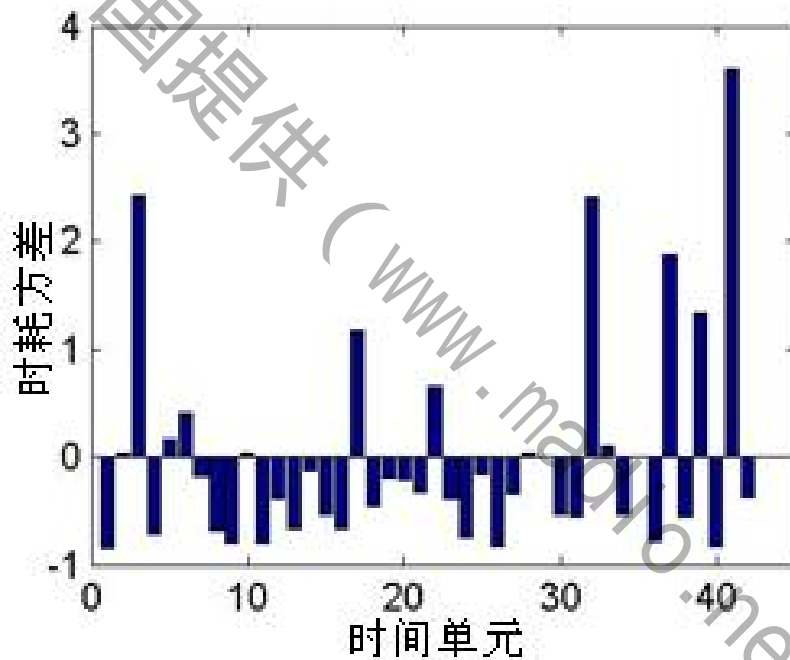


图30

在2004年7月，对应永定门桥改建，当我们持续控制业务源提高GPS使用率时，我们发现路段NASH平衡比率也持续上升，尤其当使用率升至0.34时，NASH平衡比率上升幅度最大化。同时算法也兼顾了对时耗方差的控制。

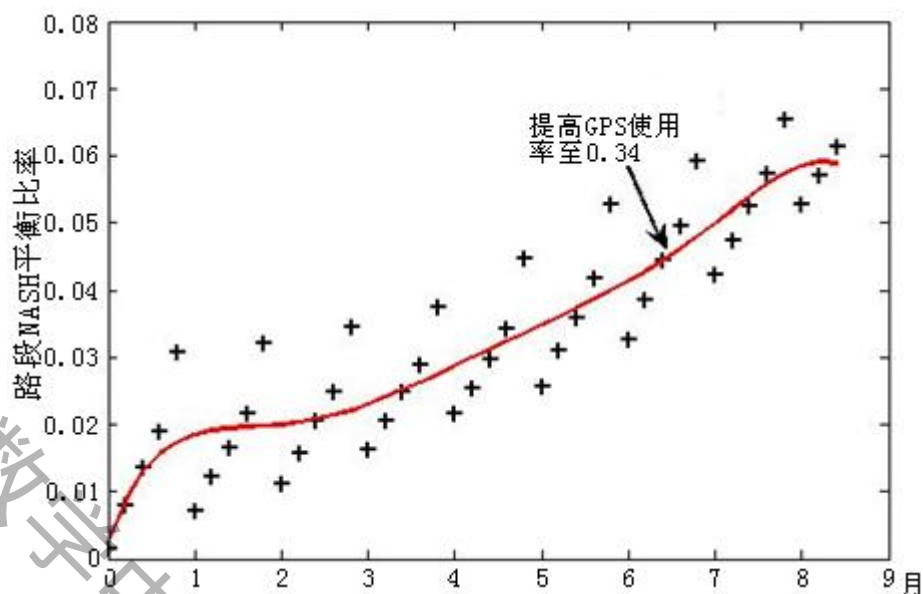


图31 2008年GPS影响下路段NASH平衡比率

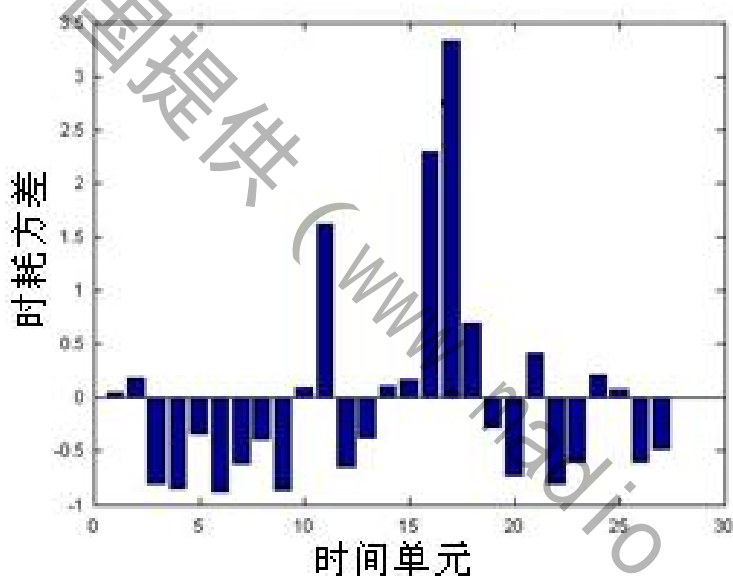


图32

在 2008 年 7 月，对应北京二环天坛路改造，当我们持续控制业务源提高 GPS 使用率时，我们发现路段 NASH 平衡比率也持续上升，尤其当使用率升至 0.34 时，NASH 平衡比率上升幅度最大化。同时算法也兼顾了对时耗方差的控制。

八、基于灰色 GOMPERTZ 模型对算法误差的分析

传统的灰色系统分析是用于控制和预测的技术。但是，从一个动态系统中得到的实际数据，往往具有一定的波动性，这种因波动性而产生的误差，会对结果产生干扰。传统的灰色模型的这种局限性实质是因为忽略了实际数据的波动性。根据改进的灰色模型——灰色 GOMPERTZ 模型将解决这一问题。该模型考虑了实际数据带有误差，以绝对误差为标准，通过背景函数导出连续算法模拟时产生的误差波动值，并形成误差波动频率并通过 MATLAB 信号处理工具箱的 CHIRP () 函数进行误差波动频率的扫频信号处理，最后利用 STEM () 函数对扫频后的信

号进行离散处理，得到精确的误差值。

灰色 GOMPERTZ 模型的实质是对以特定算法中的误差波动范围进行跟踪和预测。我们建立如下的灰色 GOMPERTZ 模型来对 ATM 路由选择的误差波动范围进行跟踪和预测（注：ATM 路由选择算法平均误差不超过 10% 为最优预测结果）：

- (1) 令 ATM 路由选择抽样次数为 $y^{(0)}$ ， $y^{(0)}(i) > 0$ ($i = 1, 2, \dots, n$) 对于 $y^{(0)}$ 采用对数变换进行生成处理，即令

$$x^{(0)}(i) = \ln[y^{(0)}(i)] \quad (i = 1, 2, \dots, n)$$

- (2) 对于数据序列 $x^{(0)}(i) = \{x^{(0)}(1), x^{(0)}(2), \dots, x^{(0)}(n)\}$ ，利用下述方程组的最小二乘解

$$\begin{bmatrix} x^{(1)}(2) + x^{(1)}(3) & 2 & 1 \\ x^{(1)}(3) + x^{(1)}(4) & 3 & 1 \\ \mathbf{M} \\ x^{(1)}(n-1) + x^{(1)}(n) & n-1 & 1 \end{bmatrix} \begin{bmatrix} p_2 \\ p_3 \\ p_4 \end{bmatrix} = \begin{bmatrix} x^{(0)}(2) \\ x^{(0)}(3) \\ \mathbf{M} \\ x^{(0)}(n-1) \end{bmatrix} - 2 \begin{bmatrix} x^{(1)}(2) \\ x^{(1)}(3) \\ \mathbf{M} \\ x^{(1)}(n-1) \end{bmatrix}$$

由 p_2 、 p_3 、 p_4 及 $\hat{y}^{(0)}(k) = e^{c+b^*e^{-a^*k}}$ 可解得参数 A；在这里， p_2 、 p_3 、 p_4 分别为概率，样本均值，分布函数值在 ATM 路由选择算法中的每个抽样模块产生的误差波动频率，通过 MATLAB 信号处理工具箱的 CHIRP（）函数进行误差波动频率的扫频信号处理得到各变量在 BOOTSTRAP 模拟过程中的频率值分别计作 C，B，K，并将结果带到公式 $\hat{y}^{(0)}(k) = e^{c+b^*e^{-a^*k}}$ 中，即可解得参数 A，进而也可再通过 MARS-AGENT 模型得到最优实例数的值。将多个模块所得到的参数 A 通过 MATLAB 信号处理工具箱的 STEM（）函数做离散处理，可得到如下的算法误差分布图：（模拟中每一时间点实例数为 5000，连续四次灰色 GOMPERTZ 模型误差波动跟踪预测结果）

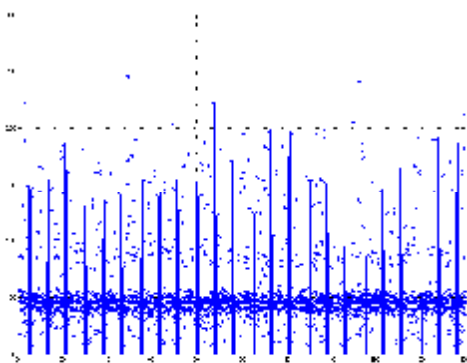


图 33

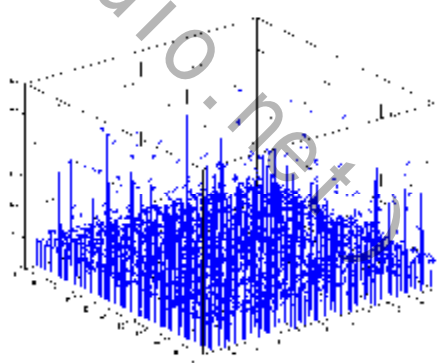


图 34

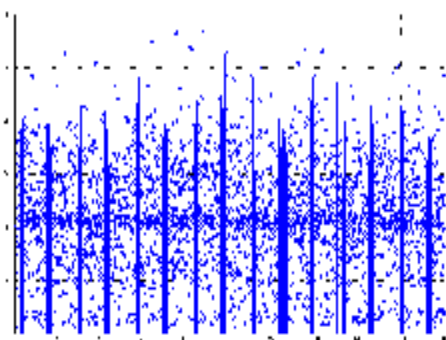


图 35

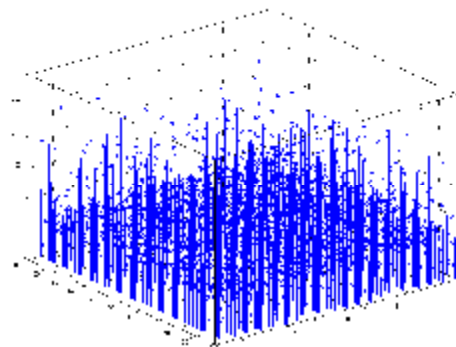


图 36

对应的 ATM 路由选择中产生的误差波动频率如下：

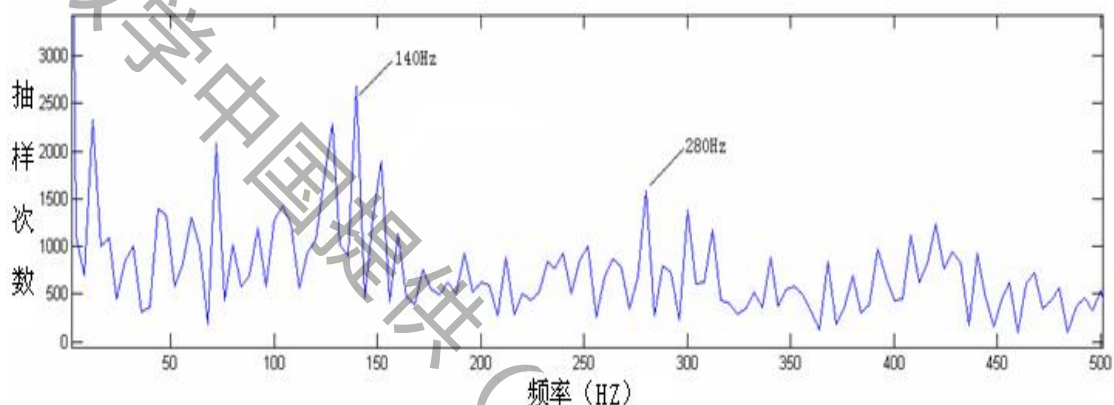


图 37 误差波动频率

结论：模拟中设置模拟实例数为5000，误差实例数292，误差范围5.84%（《10%）。

九、基于遗传算法对模型的优化检验

9.1 遗传算法流程图

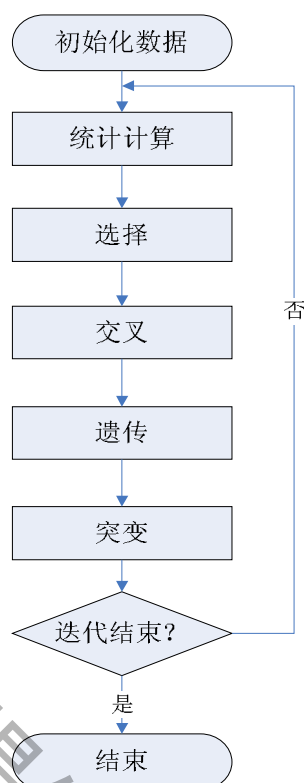


图 38 遗传算法流程图

9.2 遗传算法的实现

我们采用浮点数编码,由于浮点数编码比二进制编码在变异操作上能够保持更好的种群多样性,避免了早熟收敛(即基因丢失和基因浓度过低),减小遗传算法陷入局部最优点的概率。

9.2.1 目标函数

target_l 表示第 l 个个体的目标函数值设为

$$\text{target}_l = \sum_{i=1}^n \sum_{j=1}^m \|m_{ij} - \hat{m}(A_l, k_{l1}, k_{l2}, k_{l3}, k_{l4}, R_{ij}, T_{ij}, M_j)\|$$

遗传算法以适应度函数作为进化目标,并且只能朝着适应度函数增大的方向进化,所以需要适当地在适应度函数与目标函数之间进行转换。我们取适应度函数为目标函数的倒数,则适应度函数最大的时候即可保证目标函数最小。

9.2.2 选择与交叉

采用排序选择法,用来确定交叉个体,以及被选中的个体将产生的子代个体数量。

我们以设定选择概率

$$p_i = q'(1-q)^{r_i-1}$$

其中 p_i 为第 i 个个体的适应度函数值; Q 为选中的个体是最优个体的概率,一般选择 $Q=0.08$,种群中的个体按照适应度的大小进行排列; r_i 为种群按照个体适应

度的大小进行排序后第 l 个个体的排序号。则有

$$q' = \frac{q}{1 - (1 - q)^m}$$

其中 M 为群体中个体的总数。

为了能够在下一代产生新的个体，我们通过交叉操作来使得遗传算法的搜索能力得以飞跃地提高。变异本身只是一种局部搜索，但是与交叉结合在一起后，保证了遗传算法的有效性，进而防止了基因丢失和基因浓度过低，即早熟收敛。具体浮点数编码交叉方式为

$$c_1 = p_1 a + p_2 (1 - a)$$

$$c_2 = p_1 (1 - a) + p_2 a$$

其中， c_1 、 c_2 为个体， p_1 、 p_2 为附体， a 是 0、1 之间的随进变量。

9.2.3 变异

设置变异算子

$$X' = X + 0.5 \times L \times \Delta$$

其中， X' 为变异后变量的值， X 为变异前变量的值； L 为变量的取值范围；

$\Delta = \sum_{i=0}^m \frac{a(i)}{2^i}$ ， $a(i)$ 以 $\frac{1}{m}$ 的概率取 1，以 $1 - \frac{1}{m}$ 的概率取 0；当 $M=20$ 时， $a(i)$ 取 1

的概率为 0.05，取 0 的概率为 0.95。

9.2.4 最优保存策略：

最好的染色体不一定出现在最后一代，所以，开始时我们把最好的染色体保留下来并记作 V_0 。若在产生的新的群体中发现比 V_0 更好的染色体，则用其替代 V_0 。随着新个体的产生， V_0 不断更新，这一进程结束后，得到的 V_0 就是优化后的路段 NASH 平衡比率

9.3 优化结果及遗传状态的分析

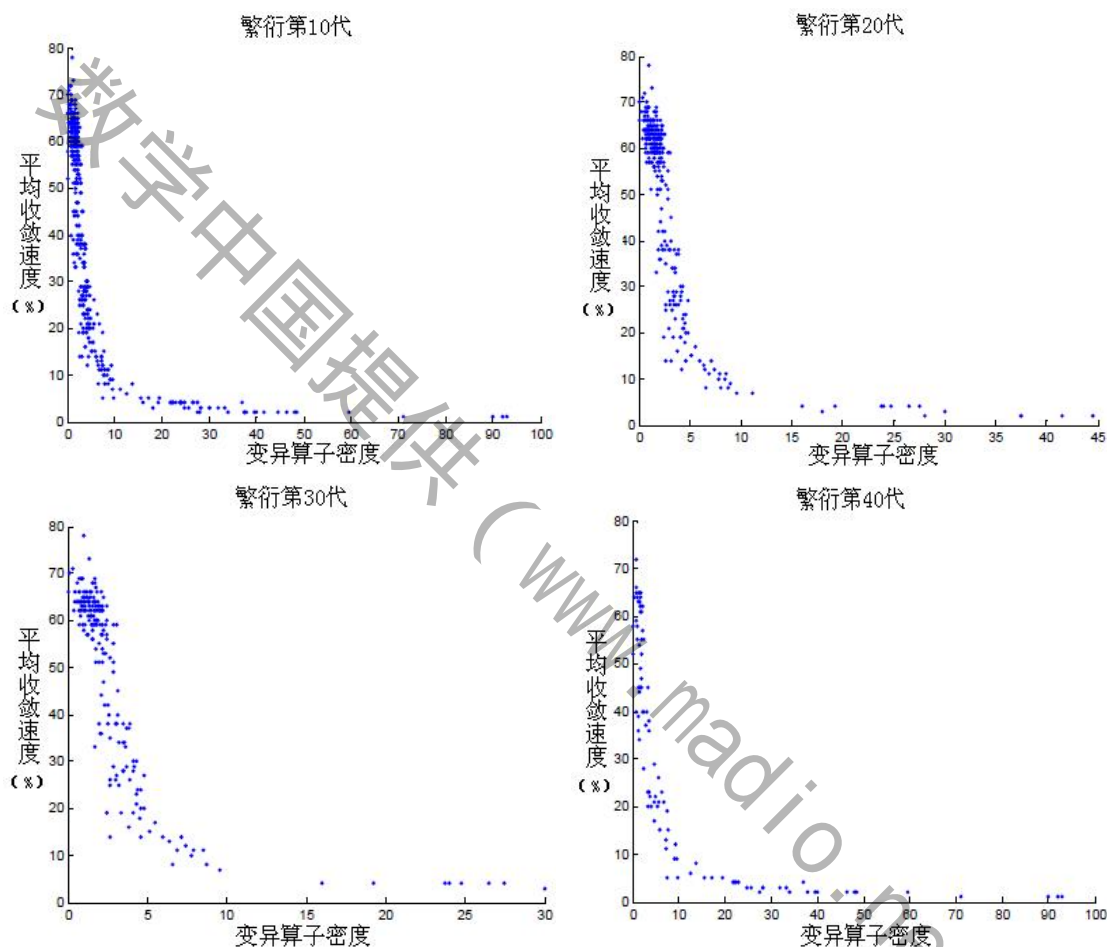
根据以上模型对路段 NASH 平衡比率进行模拟(预测实例数 10000 单元)，结果如下表：

年份	路段 NASH 平衡比率	遗传算法优化结果
2000	0.284	0.278
2001	0.831	0.826
2002	0.402	0.413
2003	0.583	0.552
2004	0.244	0.289
2005	0.912	0.898
2006	0.394	0.403
2007	0.190	0.199
2008	0.294	0.312
2009	0.238	0.209

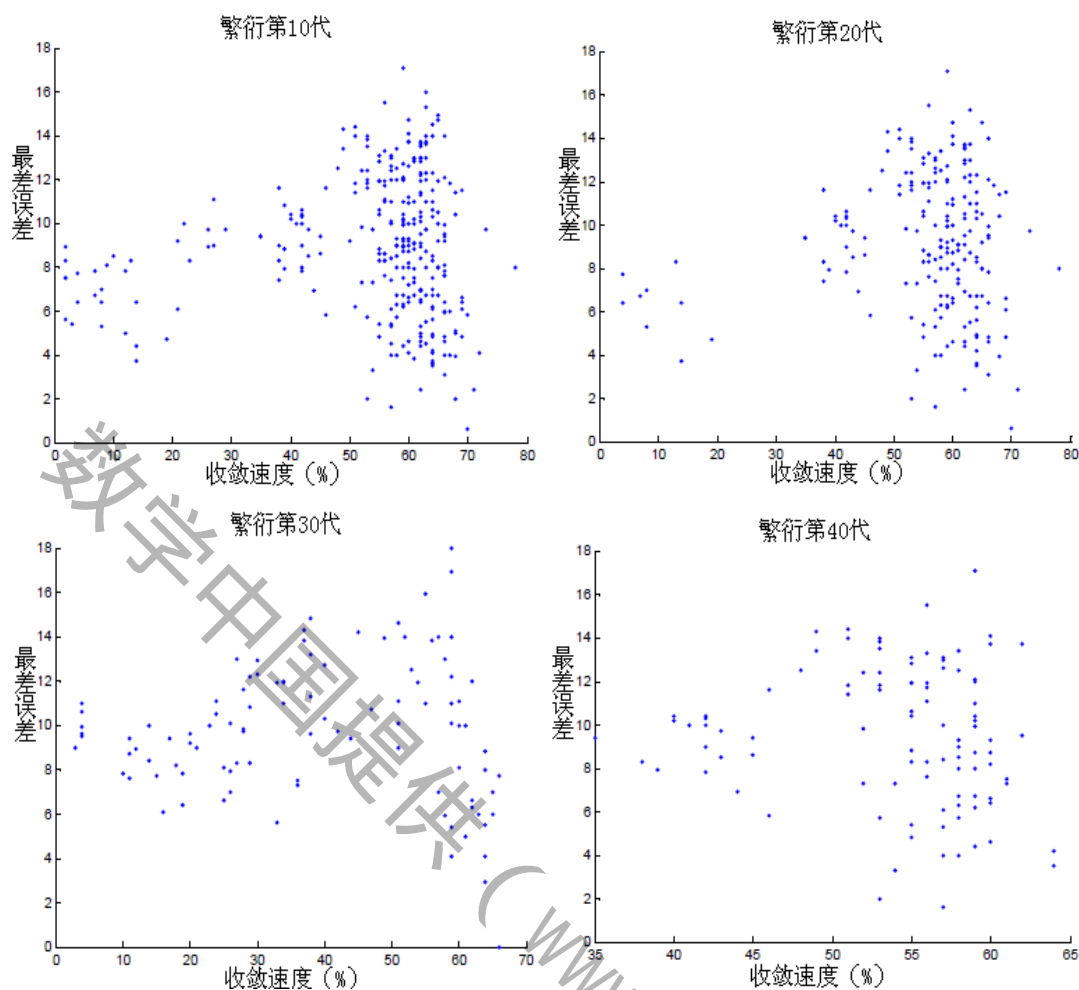
表 3 遗传算法优化结果

下图是遗传算法优化过程中个体误差的变化情况，从图 39 中可以看出，随着子代数的增加，平均个体误差逐渐下降，并最终收敛于最小误差，对应于最优个体的最小误差在经过约 40 代的遗传之后达到其最小值。而最大误差随着遗传代数增加而不断下降。这说明了遗传算法中的个体最终收敛到非线性问题的最优解。

下图为模拟过程中平均收敛速度（基于百分比）与变异算子密度的关系：



系列图 39 平均收敛速度（基于百分比）与变异算子密度关系
下图为模拟过程中最差误差与收敛速度的量化关系



系列图 40 最差误差与收敛速度的量化关系

9.4 结论:

由上述模拟图所示, 在本算法中随着子代增多, 收敛速度不断降低, 最差误差也不断下降, 在繁衍到第 40 代时, 收敛速度降低到最低, 相应的最差误差和变异算子的密度达到平衡状态, 此时的遗传算法优化的结果最佳, 同时也横向证明了 ATM 路由选择算法对 GPS 动态导航研究结果的可靠性。

参考文献

- [1]ARNOTT R, KISMAILTHE ECONOMICS OF TRAFFIC CONGESTION [J] AMERICAN SCIENTIST 82 , 1994: 446 - 4551
- [2]TBASSROAD TO RUIN [J] DISCOVER , 1992 (5) : 56 - 611
- [3]HAI YANG, M G H, BELLIA CAPACITY PARADOX IN NETWORK DESIGN AND HOW TO AVOID IT [J] TRANSPORTATION SCIENCE, 1998: 539 - 5451
- [4]C. DAGANZOQUEUE SPILLOVERS IN TRANSPORTATION NETWORKS WITH CHOICES [J] TRANSPORTATION SCIENCE, 1998: 3 - 111
- [5]周晶. 城市交通系统分析与优化 [M] 东南大学出版社, 11, 2001, 1
- [6]段里仁. 城市交通概论—交通工程学原理与应用 [M] 北京大学出版社, 1986, 11
- [7] 北京交通中心—2000~2009 年《北京交通发展年报》
- [8] 北京交通中心—2000~2009 年《全市综合交通调查》
- [9] 晏克非 ,苏永云 ,黄翔等. 车辆导航系统基于 GIS 的动态 K 最短路径递推解法[J]. 西安公路交通大学学报 ,2001 , 21(1) :64~67.

附录

1 GM(1, 1)灰色模拟源代码

% III 为预测点

% X 为预测序列（一行）

% XOX 为预测值（对有 III 而言）

% QOQ 为预测精度

% PP 为小误差概率

% C 为后验误差比

% PP>0.95 或 C<0.35 为 1 级(好)

% PP>0.80 或 C<0.50 为 2 级(合格)

% PP>0.70 或 C<0.65 为 3 级(勉强合格)

% PP<=0.70 或 C>=0.65 为 4 级(不合格)

%

% AUA=[A; U; UA; XUA];

% XEG=[XG; EG];

% PPC=[PP; C];

FUNCTION [XOX, QOQ, PPC, PDDSTR, XII, AUA, XEG]=GM11(X, III);

%%%%%%%%%

%例子

%X=[3.711 3.723 3.716 3.721 3.728]

%X=[2.67 3.13 3.25 3.36 3.56 3.72]

% X=[2.97 3.23 3.29 3.46 3.59 3.71]

%X=[43.45 47.05 52.75 57.14 62.64 68.52]

% X=[3.38 4.27 4.55 4.69 5.59]

% X=[4.24 4.33 5.20 6.42 7.32 8.53 8.82 10.72]

%

% X=[3.38 4.27 4.55 4.69 5.59]

% III=[5 6 7 8 9 10 11]

% III=[]

%%%%%%%%%

%%%%%%%%%

%% %%

%% 一、数据处理 %%

%% %%

%%%%%%%%%

%1 建立 X1 生成数列

%FORMAT LONG

A12=SIZE(X);

N=A12(2);

X1=ONES(1, N);

```

X1(1)=X(1);
FOR I=2:N
    X1(I)=X1(I-1)+X(I);
END
X1;
%2 构造数据矩阵 B 和数据向量 YN:
B=ONES(N-1, 2);
FOR I=1:N-1
    B(I, 1)=(-0.5)*(X1(I)+X1(I+1));
END
YN=ONES(N-1, 1);
FOR I=2:N
    YN(I-1, 1)=X(I);
END
B;
YN;
%3 计算 B' B, INV(B' B) 和 B' YN; 得到参数 A 和 U
%BTB=B' *B; %
%BTB=INV(BTB);
%BTYN=B' *YN;
%AU=BTB*BTYN;
AU=INV(B' *B)*B' *YN;

A=AU(1, 1);
U=AU(2, 1);

% FPRINTF(' 参数 A: %F\n', A);
% FPRINTF(' 参数 U: %F\n', U);
UA=U/A;
XUA=X(1)-UA;
%4 得出预测模型
%预测第 I 的预测值
%有问题

% XI 1=XUA*EXP((-A)*I)+UA; %

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%          二、模型检验          %%
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```
%5: 残差检验
%51 计算 X1
X1=ONES(1,N);
FOR I=1:N
    X1(I)=XUA*EXP((-A)*(I-1))+UA;
END
X1;
%52 累减生成 X0 序列
X0=ONES(1,N);
X0(1)=X1(1);
FOR I=2:N
    X0(I)=X1(I)-X1(I-1);
END
X0;
%53 计算绝对误差及相对误差序列
DD=ABS(X0-X);%计算绝对误差序列
DDD=(DD./X). *100;%计算相对误差序列 X%
% FPRINTF(' 相对误差序列: ');
% FPRINTF(' %F% ', DDD);
% FPRINTF(' \n ');
%6 进行关联度检验
%61 计算序列绝对误差 XX
XX=ONES(1,N);
FOR I=1:N
    XX(I)=X(I)-X0(I);
END
% FPRINTF(' 序列绝对误差: ');
% FPRINTF(' %F ', XX);
% FPRINTF(' \n ');

MINXX=MIN(DD);
% FPRINTF(' 最小差: ');
% FPRINTF(' %F ', MINXX);
% FPRINTF(' \n ');

MAXXX=MAX(DD);
% FPRINTF(' 最大差: ');
% FPRINTF(' %F ', MAXXX);
% FPRINTF(' \n ');
%62 计算关联系数 P=0.5 (则 R=0.6)
P=0.5;
NI=ONES(1,N);
```



```

FOR I=1:N
    NI(I)=(MI NXX+P*MAXXX)/(DD(I)+P*MAXXX);
END
%63 计算关联度, (P=0.5 时, 则 R=0.6)
R=(1/N)*SUM(NI);
% FPRINTF(' 请查找 P=0.5 是的检验准则 R=0.6 是否大于 %F\n', R);
%7 后验差检验
%71
XM=MEAN(X);
%72 求的均方差
S1=(SUM((X-XM).^2)/(N-1))^(1/2);
%73 计算残差的均值
DD=MEAN(DD);
%74 计算残差的均方差
S2=(SUM((DD-DDM).^2)/(N-1))^(1/2);
%75 计算后验误差比 C:
C=S2/S1;
% FPRINTF(' 验误差比 C: %F\n', C);
%76 计算小误差概率
%%%%%%%%%%%%%%

PR=ABS(XX-MEAN(XX))<0.6745*S1; %M 满足条件的样本
PA=SIZE(FIND(PR==1));
PS=PA(1,2); %M 满足条件的样本个数
PB=SIZE(PR);
PS=PB(1,2); %M 总样本个数

PP=PS/PS; %小误差概率
% FPRINTF(' 小误差概率 PP: %F\n', PP);

%%%%%%%%%%%%%%
%检验 预测精度 YCD

% FPRINTF(' PP>0.95 或 C<0.35 为 1 级(好)\n PP>0.80 或 C<0.50
为 2 级(合格)\n PP>0.70 或 C<0.65 为 3 级(勉强合格)\n PP<=0.70
或 C>=0.65 为 4 级(不合格)\n\n', C);
IF PP>0.95
    PD=1;
% FPRINTF(' 因 PP>0.95 且 ');
END
IF PP<=0.95&PP>0.80
    PD=2;
% FPRINTF(' 因 PP<=0.95&PP>0.80 且 ');
END

```

```
IF PP<=0.80&PP>0.70
    PD=3;
%    FPRINTF('    因 PP<=0.80&PP>0.70 且 ');
END
IF PP<=0.70
    PD=4;
%    FPRINTF('    因 PP<=0.70 且 ');
END

PD1=0;
IF C<0.35
    PD1=1;
%    FPRINTF(' C<0.35 ');
END
IF C>=0.35&C<0.50
    PD1=2;
%    FPRINTF(' C>=0.35&C<0.50 ');
END
IF C>=0.50&C<0.65
    PD1=3;
%    FPRINTF(' C>=0.50&C<0.65 ');
END
IF C>=0.65
    PD1=4;
%    FPRINTF(' C>=0.65 ');
END

PDD=MAX(PD, PD1);

IF PDD==1
    PDDSTR='1 级（好）';
%    FPRINTF(' 故根据经验，预测精度为 1 级（好）\N\n');
END
IF PDD==2
    PDDSTR='2 级（合格）';
%    FPRINTF(' 故根据经验，预测精度为 2 级（合格）\N\n');
END
IF PDD==3
    PDDSTR='3 级（勉强合格）';
%    FPRINTF(' 故根据经验，预测精度为 3 级（勉强合格）\N\n');
END
IF PDD==4
    PDDSTR='4 级（不合格）';
%    FPRINTF(' 故根据经验，预测精度为 4 级（不合格）\N\n');
```

END

%%

%%

%%

%% 三、预测 %%

%% %%

%%

%8 模型经验合格后可用于预测，预测公式为： $X_0(I+1)=XX(I+1)-XX(I)$

A1=SIZE(III);

N1=A1(2);

X0X=ONES(1,N1);

FOR I=1:N1

X0X(I)=XUA*(EXP((-A)*(III(I)))-EXP((-A)*(III(I)-1)));

% FPRINTF('预测结果：第%D个值(原数据的第%D个序列号)的预测结果为：

%F\N',III(I),III(I)+1,X0X(I));

END

PX=[1:N];

% PLOT(PX,X,'-K',III+1,X0X,'-R');

%%

%%

%%

%% 三、预测值精度评估 %%

%% %%

%%

%计算出的模型值为

XG=ONES(1,N);

XG(1)=X(1);

FOR I=2:N

XG(I)=XUA*(EXP((-A)*(I-1))-EXP((-A)*(I-2)));

END

% FPRINTF('计算出的模型值为：');

% FPRINTF('%F ',XG);

% FPRINTF('\N');

%残差分别为

EG=X-XG;

% FPRINTF('残差分别为：');

% FPRINTF('%F ',EG);

% FPRINTF('\N');

```

Q=INV(B' *B);
Q11=Q(1, 1);
Q12=Q(1, 2);
Q21=Q(1, 2);
Q22=Q(2, 2);

Q0=((EG*EG')/(N-1))^0.5;

Q0Q=ONES(1, N);
FOR K=0: N-1

Q0Q(K+1)=((A*K*X(1)-X(1)-K*U)^2*Q11+Q22+2*(A*K*X(1)-X(1)-K*U)*Q12)^0.5*EXP(-A*K)*Q0;
END
% FPRINTF(' \N' );
% FOR I=1: N
% FPRINTF(' 原数据的第%D 个序列号的预测值为: %F±%F\N', I, X(I), Q0Q(I));
% END
% FPRINTF(' \N' );
%=====
AUA=[A; U; UA; XUA];
XEG=[XG; EG];
PPC=[PP; C];
%=====
%所以预测值为
XII=0;
IF ISEMPY(III)
    XII=' -';
    X0X=' -';
    Q0Q=' -';
    RETURN;
END
FPRINTF(' ');
Q0Q=ONES(1, N1);
FOR K=1: N1

Q0Q(K)=((A*III(K)*X(1)-X(1)-III(K)*U)^2*Q11+Q22+2*(A*III(K)*X(1)-X(1)-III(K)*U)*Q12)^0.5*EXP(-A*III(K))*Q0;
% FPRINTF(' 预测结果: 第%D 个值(原数据的第%D 个序列号)的预测值为: %F±%F\N', III(K), III(K)+1, X0X(K), Q0Q(K));
END

```

四 遗传算法源代码

说明：FGA.M 为遗传算法的主程序；采用二进制GRAY编码，采用基于轮盘赌法的非线性排名选择，均匀交叉，变异操作，而且还引入了倒位操作！

```
function
[BestPop, Trace]=fga(FUN, LB, UB, er anum, popsi ze, pCross, pMutati on, pl nvers
ion, options)
% [BestPop, Trace]=fmaxga(FUN, LB, UB, er anum, popsi ze, pcross, pmutati on)
% Finds a maximum of a function of several variables.
% fmaxga solves problems of the form:
%      max F(X) subject to: LB <= X <= UB
% BestPop      - 最优的群体即为最优的染色体群
% Trace        - 最佳染色体所对应的目标函数值
% FUN          - 目标函数
% LB           - 自变量下限
% UB           - 自变量上限
% er anum      - 种群的代数, 取 100--1000(默认 200)
% popsi ze     - 每一代种群的规模; 此可取 50--200(默认 100)
% pCross       - 交叉概率, 一般取 0.5--0.85 之间较好(默认 0.8)
% pmutati on   - 初始变异概率, 一般取 0.05-0.2 之间较好(默认 0.1)
% pl nversi on - 倒位概率, 一般取 0.05-0.3 之间较好(默认 0.2)
% options      - 1*2 矩阵, options(1)=0 二进制编码(默认 0), option(1)~=0
%              十进制编码, option(2)设定求解精度(默认 1e-4)
-----

T1=clock;
if nargin<3, error('FMAXGA requires at least three input arguments'); end
if nargin==3,
er anum=200; popsi ze=100; pCross=0.8; pMutati on=0.1; pl nversi on=0.15; opti o
ns=[0 1e-4]; end
if nargin==4,
popsi ze=100; pCross=0.8; pMutati on=0.1; pl nversi on=0.15; opti ons=[0
1e-4]; end
if nargin==5, pCross=0.8; pMutati on=0.1; pl nversi on=0.15; opti ons=[0
1e-4]; end
if nargin==6, pMutati on=0.1; pl nversi on=0.15; opti ons=[0 1e-4]; end
if nargin==7, pl nversi on=0.15; opti ons=[0 1e-4]; end
if find((LB-UB)>0)
error('数据输入错误, 请重新输入(LB<UB):');
end
s=sprintf('程序运行需要约%.4f 秒钟时间, 请稍
等.....', (er anum*popsi ze/1000));
di sp(s);

global m n NewPop children1 children2 VarNum
```

```

bounds=[LB; UB]'; bi ts=[]; VarNum=si ze(bounds, 1);
preci si on=options(2); %由求解精度确定二进制编码长度
bi ts=cei l( log2((bounds(:, 2)-bounds(:, 1)))' ./ preci si on)); %由设定精度划
分区间
[Pop]=Ini tPopGray(popsi ze, bi ts); %初始化种群
[m, n]=si ze(Pop);
NewPop=zeros(m, n);
chi ldren1=zeros(1, n);
chi ldren2=zeros(1, n);
pm0=pMutati on;
BestPop=zeros(eranum, n); %分配初始解空间 BestPop, Trace
Trace=zeros(eranum, length(bi ts)+1);
i=1;
while i<=eranum
    for j=1:m
        val ue(j)=feval (FUN(1, :), (b2f(Pop(j, :), bounds, bi ts))); %计算适
    应度
    end
    [MaxVal ue, I ndex]=max(val ue);
    BestPop(i, :)=Pop(I ndex, :);
    Trace(i, 1)=MaxVal ue;
    Trace(i, (2: length(bi ts)+1))=b2f(BestPop(i, :), bounds, bi ts);
    [sel ectpop]=Nonl i nearRankSel ect(FUN, Pop, bounds, bi ts); %非线性排名
    选择
    [CrossOverPop]=CrossOver(sel ectpop, pCross, round(uni drnd(eranum-i)/era
    num));
    %采用多点交叉和均匀交叉，且逐步增大均匀交叉的概率
    %round(uni drnd(eranum-i)/eranum)
    [Mutati onPop]=Mutati on(CrossOverPop, pMutati on, VarNum); %变异
    [I nversi onPop]=I nversi on(Mutati onPop, pl nversi on); %倒位
    Pop=I nversi onPop; %更新
    pMutati on=pm0+(i^4)*(pCross/3-pm0)/(eranum^4);
    %随着种群向前进化，逐步增大变异率至 1/2 交叉率
    p(i)=pMutati on;
    i=i+1;
end
t=1: eranum;
plot(t, Trace(:, 1)');
ti tle(' 函数优化的遗传算法'); xl abel (' 进化世代数(eranum)'); yl abel (' 每一
代最优适应度(maxfi tness)');
[MaxFval, I]=max(Trace(:, 1));
X=Trace(I, (2: length(bi ts)+1));
hold on; plot(I, MaxFval, '*');
text(I+5, MaxFval, [' FMAX=' num2str(MaxFval)]);

```

```

str1=sprintf('进化到 %d 代,自变量为 %s 时,得本次求解的最优值 %f\n 对应染色体是: %s',l,num2str(X),MaxFval,num2str(BestPop(l,:)));
disp(str1);
%figure(2);plot(t,p);%绘制变异值增大过程
T2=clock;
elapsed_time=T2-T1;
if elapsed_time(6)<0
    elapsed_time(6)=elapsed_time(6)+60;
elapsed_time(5)=elapsed_time(5)-1;
end
if elapsed_time(5)<0
    elapsed_time(5)=elapsed_time(5)+60; elapsed_time(4)=elapsed_time(4)-1;
end %像这种程序当然不考虑运行上小时啦
str2=sprintf('程序运行耗时 %d 小时 %d 分钟 %.4f 秒',elapsed_time(4),elapsed_time(5),elapsed_time(6));
disp(str2);

```

%初始化种群

%采用二进制 Gray 编码,其目的是为了克服二进制编码的 Hamming 悬崖缺点

function [initpop]=InitPopGray(popsi ze, bits)

len=sum(bits);

initpop=zeros(popsi ze,len);%The whole zero encoding individual

for i=2: popsi ze-1

pop=round(rand(1,len));

pop=mod([0 pop]+[pop 0],2);

%i=1 时,b(1)=a(1);i>1 时,b(i)=mod(a(i-1)+a(i),2)

%其中原二进制串: a(1)a(2)... a(n), Gray 串: b(1)b(2)... b(n)

initpop(i,:)=pop(1:end-1);

end

initpop(popsi ze,:)=ones(1,len);%The whole one encoding individual

%解码

function [fval] = b2f(bval, bounds, bits)

% fval - 表征各变量的十进制数

% bval - 表征各变量的二进制编码串

% bounds - 各变量的取值范围

% bits - 各变量的二进制编码长度

scale=(bounds(:,2)-bounds(:,1))'./(2.^bits-1); %The range of the variables

numV=size(bounds,1);

cs=[0 cumsum(bits)];

for i=1:numV

a=bval((cs(i)+1):cs(i+1));

```

fval(i)=sum(2.^(size(a,2)-1:-1:0).*a)*scale(i)+bounds(i,1);
end

selectprob=fit/sum(fit);%计算各个体相对适应度(0,1)
q=max(selectprob);%选择最优的概率
x=zeros(m,2);
x(:,1)=[m:-1:1]';
[y x(:,2)]=sort(selectprob);
r=q/(1-(1-q)^m);%标准分布基值
newfit(x(:,2))=r*(1-q).^(x(:,1)-1);%生成选择概率
newfit=cumsum(newfit);%计算各选择概率之和
rNums=sort(rand(m,1));
fitln=1;newln=1;
while newln<=m
    if rNums(newln)<newfit(fitln)
        selectpop(newln,:)=pop(fitln,:);
        newln=newln+1;
    else
        fitln=fitln+1;
    end
end

%交叉操作
function [NewPop]=CrossOver(OldPop,pCross,opts)
%OldPop 为父代种群，pcross 为交叉概率
global m n NewPop
r=rand(1,m);
y1=find(r<pCross);
y2=find(r>=pCross);
len=length(y1);
if len>2&mod(len,2)==1%如果用来进行交叉的染色体的条数为奇数，将其调整为偶数
    y2(length(y2)+1)=y1(len);
    y1(len)=[];
end
if length(y1)>=2
    for i=0:2:length(y1)-2
        if opts==0
            [NewPop(y1(i+1),:),NewPop(y1(i+2),:)]=EqualCrossOver(OldPop(y1(i+1),:),OldPop(y1(i+2),:));
        else
            [NewPop(y1(i+1),:),NewPop(y1(i+2),:)]=MultiPointCross(OldPop(y1(i+1),:),OldPop(y1(i+2),:));
        end
    end
end

```

```

    end
end
NewPop(y2,:)=OldPop(y2,:);

%采用均匀交叉
function [children1,children2]=EqualCrossOver(parent1,parent2)

global n children1 children2
hi decode=round(rand(1,n));%随机生成掩码
crossposi tion=find(hi decode==1);
holdposi tion=find(hi decode==0);
children1(crossposi tion)=parent1(crossposi tion);%掩码为 1，父 1 为子 1
提供基因
children1(holdposi tion)=parent2(holdposi tion);%掩码为 0，父 2 为子 1 提
供基因
children2(crossposi tion)=parent2(crossposi tion);%掩码为 1，父 2 为子 2
提供基因
children2(holdposi tion)=parent1(holdposi tion);%掩码为 0，父 1 为子 2 提
供基因

%采用多点交叉，交叉点数由变量数决定
function [Children1,Children2]=MultiPointCross(Parent1,Parent2)

global n Children1 Children2 VarNum
Children1=Parent1;
Children2=Parent2;
Points=sort(uni drnd(n,1,2*VarNum));
for i=1:VarNum
    Children1(Points(2*i-1):Points(2*i))=Parent2(Points(2*i-1):Points
(2*i));
    Children2(Points(2*i-1):Points(2*i))=Parent1(Points(2*i-1):Points
(2*i));
end

%变异操作
function [NewPop]=Mutation(OldPop,pMutation,VarNum)

global m n NewPop
r=rand(1,m);
posi tion=find(r<=pMutation);
len=length(posi tion);
if len>=1
    for i=1:len
        k=uni drnd(n,1,VarNum); %设置变异点数，一般设置 1 点
    end
end

```



```

    for j=1:length(k)
        if OldPop(position(i),k(j))==1
            OldPop(position(i),k(j))=0;
        else
            OldPop(position(i),k(j))=1;
        end
    end
end
end
NewPop=OldPop;

%倒位操作

function [NewPop]=Inversion(OldPop,pInversion)

GLOBAL M N NEWPOP
NEWPOP=OLDPOP;
R=RAND(1,M);
POPIN=FIND(R<=PINVERSION);
LEN=LENGTH(POPIN);
IF LEN>=1
    FOR I=1:LEN
        D=SORT(UNIDRND(N,1,2));
        IF D(1)~=1&D(2)~=N
            NEWPOP(POPIN(I),1:D(1)-1)=OLDPOP(POPIN(I),1:D(1)-1);
            NEWPOP(POPIN(I),D(1):D(2))=OLDPOP(POPIN(I),D(2):-1:D(1));
            NEWPOP(POPIN(I),D(2)+1:N)=OLDPOP(POPIN(I),D(2)+1:N);
        END
    END
END

```