

“互联网+”时代的出租车资源配置

摘 要

本文针对“互联网+”时代的出租车资源配置问题，以综合评价理论、概率统计和博弈论为基础，分别建立了动态加权综合评价模型、灰色预测模型和双矩阵对策模型，对不同补贴方案情况下出租车资源供求匹配程度进行了评价，并提出一种可行的补贴方案缓解“打车难”问题。

针对问题一，建立动态综合评价模型，选取乘客满意度和出租车空载率以分别反映乘客和司机两方面的供求匹配程度，对其进行动态加权求得供求匹配指数。对于不同时间的分析，选取北京市9月3日至9月9日的匹配指数为研究对象，与乘客打车需求量对比得出结论：平时时段供求匹配较好，指标均值为0.431，每天的8时、17时和21时为打车高峰期，均值分别为0.348，0.410，0.405，出现供不应求现象，双休日和节假日供不应求现象较严重，均值为0.346。对于不同空间的分析，选取全国十座主要城市，按照指标大小进行Borda排序，Borda值越大匹配程度越好，求得供求匹配程度的Borda值由高到低依次为深圳50、广州40、成都37、武汉36、西安31、北京26、南京23、杭州22、上海19、沈阳11。

针对问题二，建立灰色预测模型，通过2009到2012年无补贴方案时出租车的满载率和乘客等待时间，预测出2013年、2014年和2015年无补贴情况下的对应指标。首先，与2013年和2014年的实际指标对比：补贴方案使满载率上升1.69%，等待时间下降2.44%。然后，用北京市9月3日至9月9日的匹配指数与预测出的2015年供求匹配指数对比，发现在平时时段匹配指数比预测值大1.21%，高峰期时匹配指数基本相等。综上所述得出结论：滴滴和快的两家公司的补贴方案对缓解打车难平时时段有微小帮助，高峰时段没有帮助；对使用自己产品的乘客有微小帮助，对改善总体打车难没有帮助，即不能缓解打车难。

针对问题三，首先建立双矩阵对策模型，对分别就不合作和合作对双方的最优策略进行分析，得出需要进行补贴的结论。然后，对用户增长率进行分析，定义补贴收益率和补贴性价比对补贴方案进行评价，运用神经网络算法求解出最佳补贴额度为6.94元，最有利平台发展的补贴方案为：乘客和司机的补贴为7元；随着竞争对手补贴额度的上升，逐步增加补贴额度；补贴额度达到15元后，不再增加补贴。

最后，提出一种基于出租车—公共交通联运、需求分流的补贴方案以解决打车难问题：出租车将乘客送到附近的公交站点、地铁入口，乘客乘坐公共交通前往目的地，对于选择此方式出行的乘客予以7元补贴，给予参与的出租车司机7元的补贴。通过补贴收益率函数预测出会有64.7%的乘客接受该补贴方案，人均乘坐里程减小27.7%，相当于出租车运力增大38.3%，因此证明此补贴方案可以有效的解决打车难问题。

关键词：出租车资源配置，动态加权综合评价，灰色预测，双矩阵对策，神经网络

一、问题重述

出租车是市民出行的重要交通工具之一，“打车难”是人们关注的一个社会热点问题。随着“互联网+”时代的到来，有多家公司依托移动互联网建立了打车软件服务平台，实现了乘客与出租车司机之间的信息互通，同时推出了多种出租车的补贴方案。

请你们搜集相关数据，建立数学模型研究如下问题：

- (1) 试建立合理的指标，并分析不同时空出租车资源的“供求匹配”程度。
- (2) 分析各公司的出租车补贴方案是否对“缓解打车难”有帮助？
- (3) 如果要创建一个新的打车软件服务平台，你们将设计什么样的补贴方案，并论证其合理性。

二、问题分析

该问题的中心问题是围绕出租车补贴方案，对出租车资源配置做出评估，并提出出租车补贴方案的最优解。

对于问题一，要求建立指标评估出租车资源的供求匹配程度。要分析供求匹配程度，就要分别从供和求，即司机与乘客两方面进行分析。司机方面，可以认为短时间内出租车总运力保持不变，而乘客需求量越大，司机的工作量就越大，社会资源得到充分利用。反应到指标上就是出租车的空载率，在其他因素不变时，需求量越大，空载率越低，供求匹配程度越好。

对于乘客方面，如果不考虑乘客退单的情况，可以定义乘客满意度来反映乘客需求得到满足的程度，其值与乘客等待时间成负相关。当乘客需求过大时，等待时间变长，乘客满意度下降，供求匹配程度随之下降。这样可以建立动态加权综合评价模型，整合出一个指标体现出出租车资源的供求匹配程度。

从时间上分析，可以选取北京作为代表，分析北京一周时间内每小时的供求匹配程度指标，分析每天的指标变化、工作日、双休日和节假日的指标变化。从空间上分析，可以选取全国十座主要城市，能够有效代表全国各个地区，并使用 Borda 函数方法对其匹配程度进行排名。

对于问题二，需要研究补贴方案对于缓解打车难的影响，体现在补贴是否提高了出租车供求匹配程度。为此，可以查找全国十座主要城市 2009 年至 2012 年的出租车空载率和平均乘客等待时间，建立灰色 GM(1,1) 预测模型，即可求出没有补贴情况下 2013 年、2014 年和 2015 年的相应指标，与根据实际数据计算出的 2013 年和 2014 年的相应指标对比。然后取北京 2015 年预测的供求匹配程度与一周内每小时的供求匹配程度指标对比，即可分析补贴能不能缓解打车难。

对于问题三，要求向新的打车软件服务平台提供补贴方案。首先，可以对打车软件两大巨头滴滴打车和快的打车的补贴竞争过程进行研究，建立双矩阵决策模型，分别就不合作和合作对双方的最优策略进行分析，结合实际情况对模型进行检验。如果得出的结果是需要进行补贴以获取最大利益，就需要对补贴方案进行评价。考虑到如果不进行补贴，用户增长量不变，增长率符合自然衰减函数。如果对未进行补贴时增长率的进行拟合，即可得到衰减函数。易求得每次补贴政策带来的额外增长率，对其进行积分即可求得补贴的收益率，收益率比上补贴额度即为补贴性价比。综合考虑收益率和性价比就能给出一个最有利于新平台发展的补贴方案。

最后可以结合问题二的结论，尝试提出一种新的补贴方案，以优化社会资源

配置,解决打车难的问题。可以通过改变补贴发放条件,鼓励乘客使用公共交通,实现客流量的分流,减轻出租车资源的压力,从而缓解打车难问题。再运用求得的收益率函数预测该补贴方案对出租车资源配置产生的影响,判断是否解决打车难问题。

三、问题假设

- (1) 出租车资源符合理想的交通流循环;
- (2) 乘客对等待时间的容忍足够大;
- (3) 司机和乘客只会同时使用一款打车软件;
- (4) 新创建的打车软件服务平台资本足够;
- (5) 统计数据真实可靠。

四、符号说明

W : 乘客满意度;
 σ : 出租车空车率;
 T : 乘客等待总时间;
 T_A : 出租车司机抢单时间;
 T_B : 抢单后赶到乘客所在地的时间;
 Q : 乘客打车需求人次;
 D : 出租车资源的供求匹配程度评价指标;
 η : 补贴收益率;
 k : 补贴性价比;
 L : 乘客平均乘坐里程数;
 N : 乘客打车需求量。

五、模型的建立与求解

5.1 问题一的模型建立与求解

针对问题一,需要研究不同时空出租车资源的“供求匹配”程度。首先,定义两个指标:乘客满意度 W 和出租车空车率 σ ,以分别体现乘客与司机两方面的供求匹配程度。然后,建立动态加权综合评价模型^[1],求得供求匹配综合评价指标 D 。最后,对北京 7 天内每小时的指标进行计算,分析得出时间上供求匹配程度变化规律;对全国 10 个主要城市 7 天内每天的指标进行计算,运用 Borda 函数方法得出其供求匹配程度排名。

5.1.1 问题一模型的建立

(1) 指标数据的确定与标准化处理

根据实际数据^[2],建立供求匹配程度的动态加权综合评价模型。需求量从零开始增大时,司机方面空车率下降,工作效率上升,供求匹配程度随之上升;当需求量过大时,出现“供不应求”,乘客方面供求匹配程度下降。因此,选取乘客满意度 W 和出租车空车率 σ 体现出租车资源的“供求匹配”程度。

对于乘客满意度 W ,在其他影响因素不变的情况下, W 与等待时间 T 成负相关。其中

$$T=T_A+T_B,$$

T_A 为出租车司机抢单时间; T_B 为抢单后赶到乘客所在地的时间,在此选取其为

固定值 5 min。

因此定义乘客满意度

$$W = \frac{1}{T_A + T_B}.$$

这是一个极小型指标，对其进行极差变换将其数据标准化，即令

$$W' = \frac{W - \min\{W\}}{\max\{W\} - \min\{W\}}.$$

同理，对于空车率 σ ，其与乘客打车需求人次成负相关，故定义

$$\sigma = \frac{1}{Q},$$

其中 Q 为某一时刻乘客打车需求人次。对 σ 进行标准化得

$$\sigma' = \frac{\sigma - \min\{\sigma\}}{\max\{\sigma\} - \min\{\sigma\}}.$$

(2) 动态加权函数的确定

根据实际数据分别分析乘客满意度 W 和出租车空车率 σ 对供求匹配程度的影响，选取动态加权函数为偏大型正态分布函数。其表达式为

$$A_j(x) = \begin{cases} 0, & x \leq \alpha_j, \\ 1 - e^{-\left(\frac{x - \alpha_j}{\sigma_j}\right)}, & x > \alpha_j. \end{cases}$$

(3) 综合评价指标函数的确定

考虑到实际情况下出租车数目恒定，且乘客需求量大于出租车运力，故可认为乘客满意度对供求匹配程度影响较大，取综合影响权值为 0.7，而出租车空车率的影响权值取 0.3。根据模型，求匹配综合评价指标 D 为

$$D = 0.7A_1(W)W + 0.3A_2(\sigma)(1 - \sigma).$$

匹配综合评价指标 D 越大，说明供求匹配程度越好；匹配综合评价指标 D 越小，说明供求匹配程度越差。

5.1.2 问题一模型的求解

(1) 不同时间供求匹配程度分析

根据北京 9 月 3 日到 9 月 9 日的统计数据，经计算可得北京 7 天内每小时的供求匹配程度指标如表 1 所示。

表 1 北京 9 月 3 日到 9 月 9 日供求匹配程度指标

钟点/日期	9 月 3 日	9 月 4 日	9 月 5 日	9 月 6 日	9 月 7 日	9 月 8 日	9 月 9 日
0 时	0.386	0.874	0.319	0.344	0.136	0.128	0.211
1 时	0.207	0.869	0.282	0.143	0.048	0.056	0.086
2 时	0.153	0.701	0.143	0.189	0.090	0.185	0.199
3 时	0.106	0.651	0.114	0.180	0.136	0.093	0.191
4 时	0.157	0.654	0.165	0.196	0.079	0.189	0.125
5 时	0.330	0.866	0.292	0.219	0.045	0.086	0.159

续表

6 时	0.391	0.886	0.372	0.354	0.396	0.384	0.372
7 时	0.266	0.433	0.370	0.410	0.311	0.359	0.288
8 时	0.345	0.298	0.413	0.409	0.331	0.410	0.408
9 时	0.392	0.458	0.291	0.286	0.374	0.289	0.353
10 时	0.350	0.898	0.322	0.498	0.444	0.390	0.381
11 时	0.360	0.945	0.306	0.296	0.478	0.353	0.599
12 时	0.431	0.918	0.393	0.264	0.331	0.310	0.280
13 时	0.427	0.769	0.404	0.320	0.278	0.270	0.274
14 时	0.375	0.795	0.342	0.270	0.267	0.272	0.283
15 时	0.415	0.850	0.385	0.282	0.347	0.302	0.267
16 时	0.410	0.891	0.297	0.263	0.319	0.294	0.399
17 时	0.415	0.774	0.365	0.410	0.299	0.391	0.335
18 时	0.411	0.621	0.418	0.419	0.417	0.411	0.413
19 时	0.419	0.824	0.349	0.407	0.413	0.391	0.413
20 时	0.408	0.819	0.352	0.389	0.282	0.303	0.283
21 时	0.336	0.345	0.396	0.285	0.287	0.362	0.369
22 时	0.414	0.300	0.279	0.336	0.327	0.358	0.307
23 时	0.407	0.762	0.265	0.263	0.233	0.243	0.234

将 7 天内的匹配程度指标曲线（虚线）和乘客打车需求人次曲线（实线）按比例缩放后进行对比，如图 1 所示。

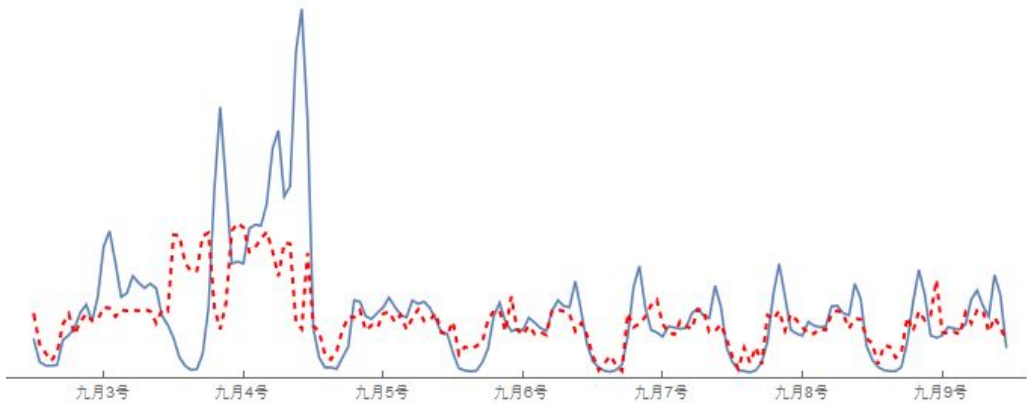


图 1 北京 9 月 3 日到 9 月 9 日供求匹配程度指标与乘客打车需求人次曲线

分析上图可知：首先，一天之内有三个需求高峰，分别在 8 点、17 点和 21 点，供求匹配指标均值分别为 0.348，0.410，0.405；平时时段指标均值为 0.431；双休日时全天需求量有所增加 0.346，指标均值为；法定节假日时（如图中 9 月 3 日、9 月 4 日阅兵放假）全天需求量有明显上升，指标均值为 0.717；工作日指标均值为 0.298。

当乘客需求人次过低时，出租车空载率上升，出现“供大于求”，此时供求匹配程度较低；乘客需求人次上升时，供求达成平衡，供求匹配程度随之上升，因此匹配程度指标与乘客打车需求人次大部分时间成正相关。但当需求人次过大时，乘客等待时间明显上升，出现“供不应求”，供求匹配程度反而下降，如法定节假日和 8 点、17 点和 21 点的需求高峰时，匹配程度指标与乘客打车需求人次成负相关，说明此时出租车运力较乘客人数有明显不足。

故得出结论，平时时段出租车资源供求匹配程度较高，但当法定节假日或每

天的需求高峰时，匹配程度较低，出现供不应求。

(3) 不同空间供求匹配程度分析

选取全国 10 座主要城市：根据 9 月 4 日到 9 月 10 日的统计数据，分别计算出每个城市一天的 D 值，如表 2 所示。

表 2 全国十座主要城市 9 月 4 日到 9 月 10 日供求匹配程度指标

日期/ 城市	北京	成都	广州	杭州	南京	上海	沈阳	深圳	武汉	西安
9 月 4 日	0.299	0.998	0.299	0.998	0.998	0.998	0.998	0.299	0.437	0.998
9 月 5 日	0.747	0.098	0.827	0.025	0.040	0.048	0.080	0.736	0.117	0.052
9 月 6 日	0.036	0.164	0.815	0.089	0.101	0.117	0.095	0.612	0.040	0.166
9 月 7 日	0.078	0.171	0.180	0.118	0.102	0.142	0.061	0.654	0.149	0.134
9 月 8 日	0.148	0.248	0.118	0.112	0.146	0.057	0.042	0.667	0.161	0.161
9 月 9 日	0.157	0.101	0.104	0.147	0.124	0.010	0.063	0.558	0.085	0.010
9 月 10 日	0.142	0.133	0.156	0.147	0.131	0.049	0.010	0.525	0.904	0.205

由 10 个城市 7 天的供求匹配程度评价指标，根据其大小进行排序，数值越大，说明匹配程度越好。由此可得反应 10 个城市匹配程度优劣的 7 个排序结果。运用 Borda 函数方法确定出一个总排序方案，则第 i 个城市 D_i 的 Borda 数为

$$B(D_i) = \sum_{j=1}^{10} B_j(s_i) \quad (i = 1, 2, \dots, 10).$$

经计算可得到各城市的 Borda 数和总排序结果如表 3 所示。

表 3 各城市的供求关系评价指标总排序结果

日期/城市	9 月 4 日	9 月 5 日	9 月 6 日	9 月 7 日	9 月 8 日	9 月 9 日	9 月 10 日	Borda 值总计
深圳	0	7	8	9	9	9	8	50
广州	0	9	9	8	3	5	6	40
成都	4	5	6	7	8	4	3	37
武汉	3	6	2	6	7	3	9	36
西安	4	3	7	4	6	0	7	31
北京	0	8	0	1	5	8	4	26
南京	4	1	4	2	4	6	2	23
杭州	4	0	1	3	2	7	5	22
上海	4	2	5	5	1	1	1	19
沈阳	4	4	3	0	0	2	0	13

得出这十座主要城市供求匹配程度排名由优到劣依次为深圳、广州、成都、武汉、西安、北京、南京、杭州、上海、沈阳。分析可以发现，大都市由于乘客需求大，出租车运力明显不足。

5.2 问题二的模型建立与求解

针对问题二，补贴方案对于缓解打车难的影响体现在补贴是否提高了出租车供求匹配程度。建立灰色 GM(1,1) 预测模型，根据全国十座主要城市 2009 年至 2012 年的出租车空载率和平均乘客等待时间，预测没有补贴情况下 2013 年、2014 年和 2015 年的相应指标：与实际数据的 2013 年和 2014 年的相应指标对比，发现补贴对空载率和乘客等待时间影响很小；与北京一周以来每天的供求匹配程度对比，发展高峰时期仍有打车难现象。综上，说明各公司的补贴方案对缓解打车难平时时段有微小帮助，高峰时段没有帮助。

5.2.1 问题二模型的建立

建立灰色 GM(1,1) 预测模型，以 2009~2012 年每年各大城市的年度平均载客率和人均等待时间构成基本序列，记为 $x_{0i}=(x_{0i}(1), x_{0i}(2), \dots, x_{0i}(4))$ ，其一次累加序列 $x_{1i}=(x_{1i}(1), x_{1i}(2), \dots, x_{1i}(4))$ ，其中

$$x_{1i}(k) = \sum_{j=1}^k x_{0i}(j), (k=1,2,\dots,4)$$

则 GM(1,1) 模型的白化微分方程为

$$\frac{dx_{1i}}{dt} + a_i x_{1i} = b_i$$

其中 a 是发展灰度， b 是内生控制发展灰度。方程的离散解为

$$x_{1i}(k) = (x_{1i}(0) - \frac{b_i}{a_i})e^{-a_i(k-1)} + \frac{b_i}{a_i} (k=2,3,4)$$

$$u_i = (a_i, b_i)^T, B_i = \begin{bmatrix} -\frac{1}{2}(x_{1i}(2) + x_{1i}(1)) & 1 \\ -\frac{1}{2}(x_{1i}(3) + x_{1i}(2)) & 1 \\ -\frac{1}{2}(x_{1i}(4) + x_{1i}(3)) & 1 \end{bmatrix}, Y_i = \begin{bmatrix} x_{0i}(2) \\ x_{0i}(3) \\ x_{0i}(4) \end{bmatrix},$$

由最小二乘法可求得使

$$J(\hat{u}_i) = (Y_i - B_i \cdot \hat{u}_i)^T (Y_i - B_i \cdot \hat{u}_i)$$

达到最小值

$$u_i = (a_i, b_i)^T = (B_i^T \cdot B_i^{-1}) B_i^T \cdot Y_i$$

于是解得

$$u_i = (a_i, b_i)^T$$

于是

$$\hat{x}_{li}(k+1) = (x_{li}(1) - \frac{\hat{b}_i}{\hat{a}_i})e^{-\hat{a}_i(k-1)} + \frac{\hat{b}_i}{\hat{a}_i}.$$

再由

$$\hat{x}_{0i}(k+1) = \hat{x}_{li}(k+1) - \hat{x}_{li}(k)(i=1,2,\dots,10)$$

即可求得年度平均载客率和人均等待时间的预测值。

5.2.2 问题二模型的求解

(1) 补贴的总体影响程度

打车软件补贴始于 2013 年，因此根据 2009 到 2012 年的统计数据，对 2013 和 2014 年的年度平均载客率和人均等待时间按照上述模型进行灰色预测，并将其与实际数据进行对比，求得差额如表 4、表 5 所示。

表 4 2013 年年度平均载客率和人均等待时间误差分析

城市	载客率预测值(%)	载客率实际值(%)	差额	等待时间预测值(min)	等待时间实际值(min)	差额
北京	67.33	69.43	3.13%	9.47	9.23	2.49%
上海	73.66	76.13	3.35%	9.83	9.57	2.67%
沈阳	60.00	60.44	0.74%	6.57	6.43	2.14%
广州	70.71	71.75	1.46%	11.27	10.65	5.54%
成都	69.70	71.42	2.46%	7.94	7.75	2.42%
杭州	69.33	69.67	0.50%	12.00	11.76	1.98%
南京	69.35	69.58	0.33%	8.57	8.44	1.49%
深圳	83.07	84.12	1.26%	9.39	9.12	2.82%
武汉	70.43	71.47	1.47%	11.27	10.88	3.45%
西安	62.02	62.59	0.92%	8.24	8.02	2.68%

表 5 2014 年年度平均载客率和人均等待时间误差分析

城市	载客率预测值(%)	载客率实际值(%)	差额	等待时间预测值(min)	等待时间实际值(min)	差额
北京	67.83	70.46	3.88%	9.42	9.23	1.98%
上海	73.16	77.24	5.57%	9.88	9.71	1.75%
沈阳	60.50	61.77	2.10%	6.89	6.76	1.94%
广州	69.28	70.38	1.59%	11.59	11.16	3.71%
成都	71.27	72.47	1.68%	7.71	7.56	1.91%
杭州	69.83	70.28	0.65%	11.80	11.4	3.40%
南京	68.86	69.27	0.59%	8.62	8.46	1.85%
深圳	85.73	85.64	0.10%	9.02	8.88	1.51%
武汉	68.10	69.01	1.34%	11.42	11.3	1.08%
西安	63.06	63.48	0.67%	8.05	7.89	2.00%

分析表格发现，补贴方案使得载客率略有上升，乘客等待时间有所下降，幅度分别为 1.69%和 2.44%。即各公司的补贴方案对缓解打车难只能起到极小的作用。

(2) 不同时段补贴的影响程度

接下来分析不同时段内打车软件补贴对打车难问题的影响。通过灰色预测和问题一中的综合评价模型，计算出北京 2015 年年度供求匹配程度指标

$$D_0 = 0.3019.$$

将其与北京一周以来每小时供求匹配程度指标与乘客打车需求人次对比，如图 2 所示。

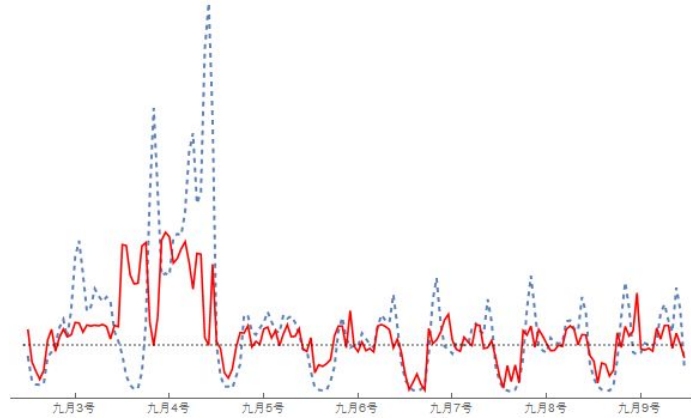


图 2 供求匹配指标比较

通过对比无补贴时年平均指标（水平线）和每小时供求匹配指标（实线）可以发现：在平时时段，有补贴影响的指标比无补贴的值大 1.21%，说明各公司的补贴方案对缓解打车难只能起到较小的作用；当需求量较大时（如 9 月 4 日的高峰时段），供求匹配指标下降至无补贴水平，说明在高峰时段打车软件补贴方案不能改善打车难问题。

综合以上求解过程，得出结论：滴滴和快的两局公司的补贴方案对缓解打车难平时时段有微小帮助，高峰时段没有帮助；对使用自己产品的乘客有微小帮助，对改善总体打车难没有帮助，即可以认为不能缓解打车难。

5.3 问题三的模型建立与求解

针对问题三，在市场拓展阶段，建立双矩阵对策模型^[1]，对现有两大打车服务平台滴滴打车和快的打车就补贴方面的竞争过程进行分析，得出新平台需要进行补贴以抢占市场的结论。针对补贴的额度，定义补贴收益率和补贴性价比来反映补贴方案的优劣程度。通过分析滴滴打车和快的打车历史搜索指数^[3]，确定数次补贴的收益情况，运用神经网络拟合求得补贴的收益曲线与性价比曲线，得出最有利于公司发展的补贴额度。

在市场稳定阶段，针对问题二中补贴不能解决打车难的问题，提出一种基于出租车与公共交通联运的补贴方案，并计算得出部分乘客采用此补贴方案时，出租车相对运力显著提升，说明改进后的补贴方案较好的解决了打车难问题。

5.3.1 问题三模型的建立

(1) 双矩阵对策模型的建立

用 S_1 和 S_2 代表打车软件市场上的两大公司，建立如表 6 所示的双矩阵对策模型。两者的策略均为“补贴”或“不补贴”。假如二者都选择不补贴，则 S_1 的收益为 R_1 ， S_2 的收益为 R_2 ，两者的博弈结果为 (R_1, R_2) 。如果其中一个选择补贴，那么它会因为提供补贴而增加用户使用数量，提高市场份额，于是收益增加 R_A ；同理没有降价的一方收益会减少 R_B ，且 $R_A < R_B$ 。如果二者都补贴，都会

因为进行补贴而增加成本，造成一定的损失 R_0 。但由统计数据可知 $R_A > R_0$ ，即无论对手是否补贴，自己进行补贴都会使收益最大化。

表 6 双矩阵对策模型

S_1/S_2	补贴	不补贴
补贴	$(R_1 - R_0, R_2 - R_0)$	$(R_1 + R_A, R_2 - R_B)$
不补贴	$(R_1 - R_B, R_2 + R_A)$	(R_1, R_2)

在完全弱意义下对该博弈模型求解，按照优越原则进行简化，求得简化对策为 $G = \{S_1^T, S_2^T; A, B\}$ ，双方赢得为 $(R_1 - R_0, R_2 - R_0)$ ，所以纳什平衡点为(补贴，补贴)。

因此，在双方没有任何形式的联合或协商，也不能互通信息，即双方是直接对抗的情况下，双方的最优策略均为进行补贴。现实中，滴滴打车和快的打车确实不断增加补贴额度以抢占市场，验证了模型的可行性。

对于双矩阵对策，不合作情况下双方只考虑个人的最优解，在全局上未必是最好的结果。为此，建立合作的双矩阵对策模型，求得不同情况下市场总收益如表 7 所示。

表 7 市场总收益

S_1/S_2	补贴	不补贴
补贴	$R_1 + R_2 - 2R_0$	$R_1 + R_2 + R_A - R_B$
不补贴	$R_1 + R_2 - R_B + R_A$	$R_1 + R_2$

显然，双方都不补贴时，市场总收益最大，(不补贴，不补贴)为模型在合作情况下的最优解。实际上，2014 年 5 月双方均停止了对乘客和司机的补贴，符合模型的结果。

(2) 补贴方案评价指标的确定

考虑到滴滴打车和快的打车与 2014 年 5 月已经停止补贴并达成合作，占据 95% 以上的市场份额。新公司若想抢占市场，可以认为符合双矩阵对策模型，需发放补贴提升市场占有率。但单位金额的补贴所产生的效益并非不变，为此定义补贴收益率 η 和补贴性价比 k 来反映补贴方案的优劣程度：

$$\eta = w - w_0, k = \frac{\eta}{S},$$

其中 w 为用户增长率， w_0 为用户增长率自然衰减函数， S 为补贴额度。

5.3.2 问题三模型的求解

(1) 市场拓展阶段补贴方案的确定

对 2014 年 1 月至 2014 年 5 月期间滴滴打车的搜索次数进行统计，即可近似得出其用户增长量，对增长量累积求和得到用户总数量如图 3 所示。

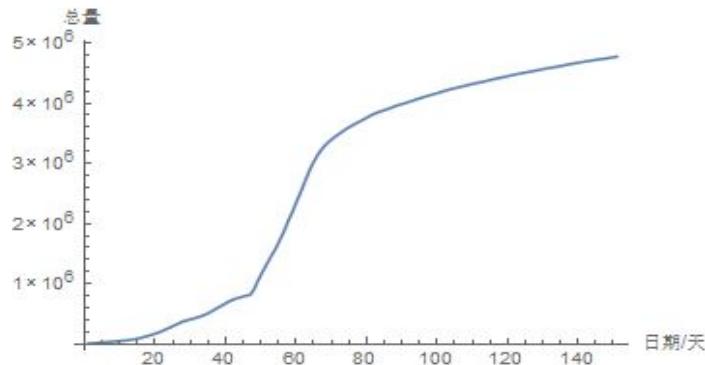


图 3 2014 年 1 月至 5 月期间滴滴打车用户总量

则用户增长率

$$w = \frac{n - \Delta n}{t - \Delta t},$$

其中 n 为用户数量, t 为日期。

对总数量求导得到其用户增长率如图 4 中实线所示。当增长量保持不变时, 随着用户总体数量的增加, 增长率成自然衰减, 选取 1 月初和 5 月末无补贴时段, 对自然衰减函数进行拟合, 结果如图 4 中虚线所示。可以看到, 在不进行补贴时, 二者符合得较好。

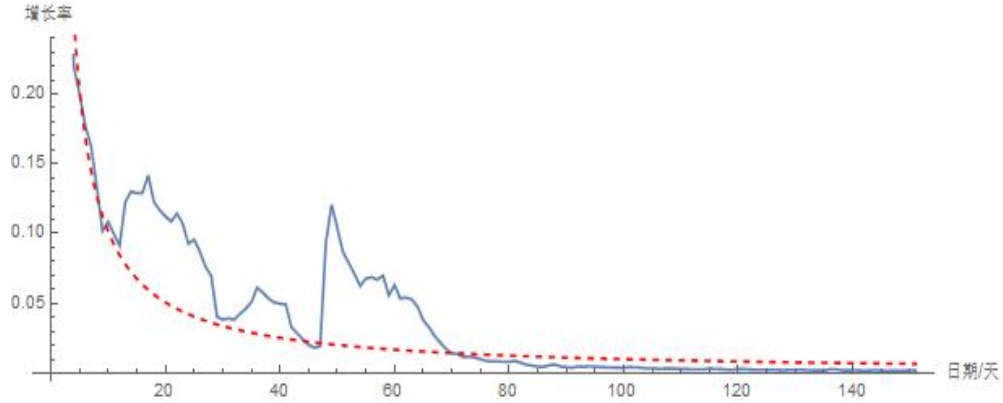


图 4 2014 年 1 月至 5 月期间滴滴打车用户增长率与自然衰减函数

结合滴滴打车的补贴政策得出: 1 月 10 日乘客补贴上调至 10 元, 2 月 18 日上调至 16 元。可以认为增长率高于自然衰减函数的部分即为补贴政策产生的收益率 η 。且在下一次补贴政策变动前, 增长率已回归自然衰减函数, 故可认为两次补贴政策收益相互独立。由定义即可求得收益率 η , 对快的打车采用同样方法计算, 得到 5 组补贴额度与收益率对应关系如下: (10, 0.926499), (16, 0.967951), (10, 0.81863), (13, 1.02049), (0, 0)。运用神经网络拟合, 求得的补贴收益率 η 和补贴性价比 k 分别如图 5、6 所示。

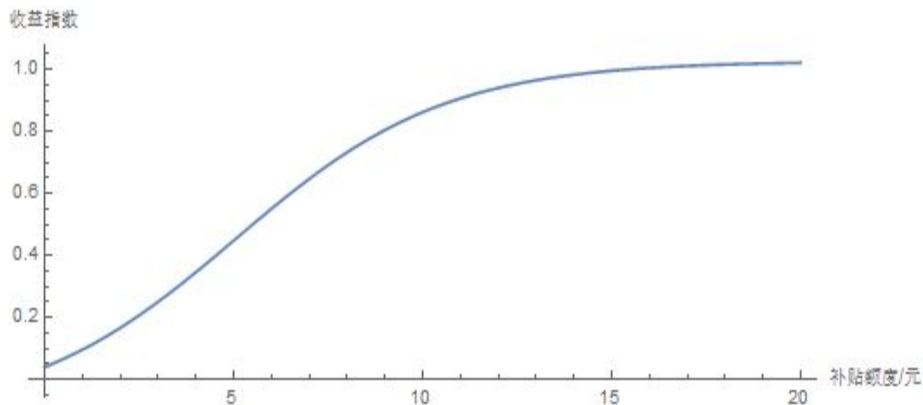


图 5 补贴收益率与补贴额度的关系

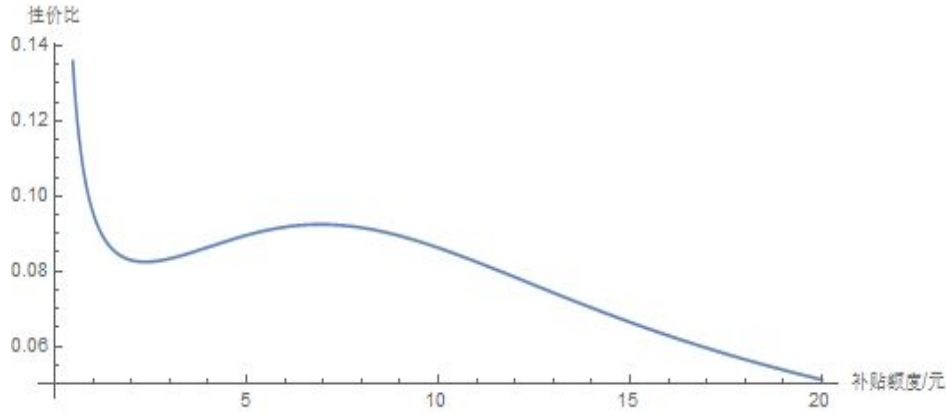


图 6 补贴性价比与补贴额度的关系

可以看出，随着补贴额度的增加，收益率随之增加。但性价比呈现先下降再上升最后不断下降的趋势。考虑到根据博弈论分析出新公司需要以补贴打开市场，因此补贴额度不能过低。

对性价比函数在补贴额度[5,10]区间内求极大值，得出补贴额度为 6.94 元时，性价比最高。当补贴额度进一步提高时，性价比也随之下落，且 15 元之后即使提高补贴也不能显著提高收益。

因此，综合双矩阵对策模型、收益指数、性价比，给新打车平台提供了一个补贴方案：首先，给乘客和司机的补贴额度为 7 元；随着竞争对手补贴额度的上升，逐步增加补贴额度；补贴额度达到 15 元后，不再增加补贴。

(2) 解决“打车难”问题的补贴方案

根据问题二的求解过程可知，各公司现行的补贴方案对打车难问题没有帮助，即只调整补贴的数额不能从根本上解决此问题。因此，市场拓展阶段求得的补贴方案需要改进。

由问题一中的供求匹配程度评价指数

$$D = 0.7A_1(W)W + 0.3A_2(\sigma)(1 - \sigma)$$

可知，从根本上解决打车难问题就要提高出租车运力、降低乘客打车需求量，而提高出租车运力不易实现。为此，可以通过补贴的方式诱导部分乘客将出行路程转移到公共交通上，变相降低乘客需求量，以缓解出租车系统的压力，具体步骤如下：

- (1) 乘客下达直达目的地的订单；
- (2) 规划一条路线，出租车会将乘客送到附近的公交站点、地铁入口，乘客乘坐公共交通前往目的地，对于选择此方式出行的乘客予以 7 元补贴；
- (3) 给予参与的出租车司机 7 元的补贴。

通过这种方式，一定程度上实现了客流量的分流，缓解了打车难的问题，但新建立的打车平台的利益得不到保证。为此，可以与政府部门合作，将平台转型为一个公益性质的出租车资源调配平台，以解决打车难的问题。

下面对该方案的效果进行验证。定义乘客打车需求总量为

$$N = L \cdot Q$$

其中 L 为乘客平均乘坐里程数。

由统计数据可得 2015 年出租车乘客乘坐里程数百分比。考虑到出租车分布地区 3 公里之内至少存在一个公共交通入口，故可以认为 3 公里以内的乘客不受

补贴影响，3 到 5 公里、10 到 15 公里、5 到 10 公里的乘客里程减小一个档位，15 公里以上的乘客不受影响。根据收益曲线（图 4），求得补贴额度为 7 元时，总量中有 64.7% 的乘客接受了该补贴方案，即 3 到 5 公里、10 到 15 公里、5 到 10 公里的乘客中有 86.5% 的乘客接受此补贴方案。

变动前后的各档位乘客百分比和乘客平均里程数如表 8 所示。

表 8 2015 年各里程乘客比例数据对比

里程	3 公里内	3 到 5 公里	5 到 10 公里	10 到 15 公里	15 公里以上	人均里程
补贴前	10.87	37.83	21.46	15.58	14.26	7.73
补贴后	43.6	23.67	16.37	2.1	14.26	5.59

通过计算得出人均里程减小了 27.7%，相当于出租车运力增大了 38.3%，因此证明此补贴方案可以有效的解决打车难问题。

六、模型的评价与改进

6.1 模型的评价

对于问题一，建立动态加权综合评价模型，将多个指标转化为一个指标，明显反映了各个城市和同一城市不同时间的供求匹配情况，同时也减小了统计误差对系统的影响。

对于问题二，抓住补贴时间，利用过去未补贴时间段的市場数据，由于市场本身存在多方调控，灰色预测模型是很好的解决方法，由此得出模拟的整个市场的市場数据，将其与实际市場数据进行对比，侧面反映出补贴对于市場的影响，也就是对打车难问题是否产生作用。

对于问题三，先建立建立双矩阵对策模型，对补贴方案进行定性分析，结合两大公司实际竞争历史进行检验。然后进行定量分析，根据影响力来确定补贴对于市場的影响，将所得数据进行神经网络拟合，得出了补贴与公司市場增长率的关系，由此得出了最佳补贴数额。之后再落实到问题，建立细致的补贴方案模型，并检验它对于市場的影响。事实上得出了该方案是合理并极有成效的。

6.2 模型的改进

对于所建立的几个模型，缺少了未知量的影响，可以多考虑几个未知量，进行灵敏度检验，确切的证实模型的优劣，对于数据的采集量可以适当的增大，得到更加精确的结果，以佐证模型的稳定性。

参考文献

- [1] 韩中庚，数学建模方法及其应用（第二版），北京，高等教育出版社，2009:171~176;
- [2] Big Data，苍穹一滴滴快的智能出行平台，<http://v.kuaidadi.com/>，2015/09/11;
- [3] 百度公司，百度指数，<http://index.baidu.com/>，2015/09/12;
- [4] 兰海洋，胡刚，袁海平，关于手机打车软件的调查与思考[J]，新西部，2014(20);
- [5] 刘思峰，蔡华，杨英杰，曹颖，灰色关联分析模型研究进展[J]，系统工程理论与实践，2013(8);
- [6] 王虎军，行业管制下分时段大城市出租车供需关系研究[M]，北京交通大学，2007,12;

- [7] Biao Leng, Heng Du, Jianyuan Wang, Li Li, Zhang Xiong, Analysis of Taxi Drivers' Behaviors Within a Battle Between Two Taxi Apps[J], IEEE Transactions on Intelligent Transportation Systems, 2015;
- [8] 顾君忠, 大数据与大数据分析[J], 软件产业与工程, 2013(4);
- [9] 杨燕, 博弈论在城市客运出租汽车价格管理中的应用研究[M],西南交通大学, 2011,11;
- [10] 李旭超, 论打车软件对乘客打车行为的 SWOT 分析[J], 现代工业经济和信息化, 2014,(5)。

附 录

附录一 图一的 Mathematica 10 代码

```
ClearAll["Global`*"]
a = {4685, 1976, 1541, 1517, 1600, 4668, 5244, 6129, 8029, 8952, 6913,
     10024, 16061, 17928, 14093, 9881, 10380, 12448, 11628, 10988,
     11520, 10953, 7613, 6473, 4896, 2553, 1493, 992, 1088, 2945, 8560,
     22997, 33041, 23649, 13972, 14204, 13993, 18240, 18708, 18568,
     21269, 27992, 30196, 22160, 23388, 39717, 45024, 31573, 6036, 2512,
     1296, 1312, 1124, 2497, 3845, 9505, 9309, 7556, 7212, 7973, 8613,
     9828, 8696, 7700, 7417, 9477, 9061, 9328, 8592, 7188, 5541, 5392,
     3017, 1192, 944, 825, 897, 1905, 3560, 7588, 9184, 6900, 5736,
     5953, 5872, 7376, 6761, 6112, 5753, 8172, 9488, 8800, 8637, 11821,
     8157, 4145, 2061, 1313, 893, 781, 988, 1652, 5132, 11100, 13669,
     8640, 5884, 5569, 5053, 6308, 6173, 6017, 6108, 7904, 8485, 7649,
     7340, 11261, 8701, 3637, 1933, 1004, 857, 696, 969, 1888, 4488,
     10200, 13932, 9968, 5892, 5408, 5169, 6857, 6388, 6213, 6288, 8756,
     8828, 7908, 7665, 11505, 9660, 3953, 2221, 1356, 965, 865, 840,
     1361, 4449, 9273, 13201, 10236, 5220, 4928, 5192, 6240, 6053, 5940,
     6552, 9620, 10692, 8829, 7516, 12592, 10128, 3821};
data1 = 1/a;
c = {58.64, 56.05, 60.31, 68.08, 62.96, 69.74, 63.33, 46.83, 53.01,
     56.02, 53.6, 53.56, 61.05, 62.97, 54.09, 58.38, 57.49, 64.16,
     57.32, 58.58, 65.0, 72.42, 60.18, 59.5, 25.98, 21.15, 21.48, 27.35,
     26.31, 23.22, 26.61, 40.27, 46.67, 39.67, 26.51, 23.89, 25.43,
     31.88, 31.04, 28.96, 27.17, 31.86, 35.93, 30.08, 30.28, 43.13,
     46.47, 32.25, 72.17, 65.92, 60.88, 66.37, 63.37, 57.48, 62.23,
     68.74, 58.19, 49.57, 51.85, 50.61, 56.22, 56.9, 52.67, 55.78,
     50.13, 53.95, 61.95, 70.75, 53.33, 57.0, 49.64, 48.48, 60.62,
     64.63, 64.07, 62.99, 62.29, 59.45, 64.24, 62.74, 63.99, 49.42,
     37.96, 43.78, 46.78, 51.63, 46.66, 44.65, 47.23, 58.3, 62.02,
     57.63, 55.88, 77.29, 71.69, 44.36, 50.46, 51.6, 50.68, 55.27, 49.9,
     49.26, 61.32, 74.78, 73.27, 54.73, 39.24, 38.39, 42.09, 45.09,
     47.44, 41.86, 42.86, 50.11, 61.21, 59.72, 48.87, 77.0, 72.66,
     48.25, 43.3, 47.1, 58.18, 54.52, 63.95, 48.77, 58.8, 70.06, 65.22,
     48.68, 40.59, 41.51, 42.88, 46.85, 45.79, 43.61, 44.02, 56.0,
     63.41, 56.29, 50.45, 70.05, 69.96, 48.37, 53.61, 54.01, 61.3,
     59.12, 53.45, 39.77, 57.17, 76.48, 65.31, 53.12, 40.66, 35.43,
     44.27, 48.6, 44.56, 45.94, 40.48, 52.1, 63.93, 63.17, 48.82, 69.45,
     74.9, 46.95};
data2 = 1/(c + 300);
Array[x, 168];
Array[p, 168];
For[k = 1, k < 169, k++,
```

```

x[k] = (data1[[k]] - Min[data1])/(Max[data1] - Min[data1]);
For[s = 1, s < 169, s++,
  p[s] = (data2[[s]] - Min[data2])/(Max[data2] - Min[data2]);
quan2 = Array[x, 168];
quan1 = Array[p, 168];
Array[t, 168];
For[r = 1, r < 169, r++,
  t[r] = 0.7 (1 - E^-((quan1[[r]] - 0.5)^2/0.197703^2)) quan1[[r]] +
    0.3 (1 - E^-((quan2[[r]] - 0.5)^2/0.197703^2)) (1 - quan2[[r]]);
X = Array[t, 168];
Show[ListLinePlot[a, PlotRange -> All,
  Ticks -> {{{13, "九月 3 号"}, {37, "九月 4 号"}, {61, "九月 5 号"}, {85,
    "九月 6 号"}, {109, "九月 7 号"}, {133, "九月 8 号"}, {157, "九月 9 号"}},
  Automatic}, Axes -> {True, False}],
ListLinePlot[20000 X,
  Ticks -> {{{13, "九月 3 号"}, {37, "九月 4 号"}, {61, "九月 5 号"}, {85,
    "九月 6 号"}, {109, "九月 7 号"}, {133, "九月 8 号"}, {157, "九月 9 号"}},
  Automatic}, PlotStyle -> {Dashed, Thick, Red}, PlotRange -> All,
  Axes -> {True, False}]]

```

附录二 问题二 2013 年等待时间的灰色预测 Mathematica 10 代码

```

Clear["Global`*"]
datasj0 = {{10.3`, 9.7`, 9.4`, 9.6`}, {10.1`, 9.6`, 9.9`,
  9.7`}, {6.1`, 5.5`, 6.3`, 6.1`}, {10.8`, 10.3`, 10.8`,
  10.9`}, {9.1`, 8.7`, 8.4`, 8.2`}, {13.3`, 12.8`, 12.,
  12.4`}, {8.9`, 8.5`, 8.3`, 8.6`}, {10.9`, 10.6`, 10.1`,
  9.8`}, {11.2`, 10.8`, 11., 11.1`}, {9.1`, 8.8`, 8.7`, 8.4`}};
datasj1 = {{10.3`, 20., 29.4`, 39.}, {10.1`, 19.7`, 29.6`,
  39.3`}, {6.1`, 11.6`, 17.9`, 24.}, {10.8`, 21.1`, 31.9`,
  42.8`}, {9.1`, 17.8`, 26.2`, 34.4`}, {13.3`, 26.1`, 38.1`,
  50.5`}, {8.9`, 17.4`, 25.7`, 34.3`}, {10.9`, 21.5`, 31.6`,
  41.4`}, {11.2`, 22., 33., 44.1`}, {9.1`, 17.9`, 26.6`, 35.}};
a = Array[x, 10];
For[n = 1, n < 11, n++,

  B1 = -0.5 (datasj1[[n, 1]] + datasj1[[n, 2]]);
  B2 = -0.5 (datasj1[[n, 2]] + datasj1[[n, 3]]);
  B3 = -0.5 (datasj1[[n, 3]] + datasj1[[n, 4]]);
  Y1 = datasj0[[n, 2]];
  Y2 = datasj0[[n, 3]];
  Y3 = datasj0[[n, 4]];
  B = {{B1, 1}, {B2, 1}, {B3, 1}};
  Y = {Y1, Y2, Y3};
  \[Beta] = Inverse[(B\[Transpose].B)].B\[Transpose].Y;

```



```

x[n] = ((datasj0[[n, 1]] - \[Beta][[2]]/\[Beta][[
1]]) E^(-4 \[Beta][[1]] ) + \[Beta][[2]]/\[Beta][[1]] -
datasj1[[n, 4]]);

```

a

附录三 问题二 2013 年满载率的灰色预测 Mathematica 10 代码

```

Clear["Global`*"]
datalv0 = {{64., 65., 68., 66.}, {70., 76., 73., 75.}, {56.,
58., 60., 59.}, {70., 75., 74., 72.}, {63., 65., 67.,
68.}, {65., 67., 70., 68.}, {67., 70., 72., 69.}, {72.,
75., 79., 80.}, {76., 78., 75., 73.}, {57., 59., 60.,
61.}};
datalv1 = {{64., 129., 197., 263.}, {70., 146., 219.,
294.}, {56., 114., 174., 233.}, {70., 145., 219.,
291.}, {63., 128., 195., 263.}, {65., 132., 202.,
270.}, {67., 137., 209., 278.}, {72., 147., 226.,
306.}, {76., 154., 229., 302.}, {57., 116., 176., 237.}};
a = Array[x, 10];
For[n = 1, n < 11, n++,

```

```

B1 = -0.5 (datalv1[[n, 1]] + datalv1[[n, 2]]);
B2 = -0.5 (datalv1[[n, 2]] + datalv1[[n, 3]]);
B3 = -0.5 (datalv1[[n, 3]] + datalv1[[n, 4]]);
Y1 = datalv0[[n, 2]];
Y2 = datalv0[[n, 3]];
Y3 = datalv0[[n, 4]];
B = {{B1, 1}, {B2, 1}, {B3, 1}};
Y = {Y1, Y2, Y3};
\[Beta] = Inverse[(B\[Transpose].B)].B\[Transpose].Y;
x[n] = ((datalv0[[n, 1]] - \[Beta][[2]]/\[Beta][[
1]]) E^(-4 \[Beta][[1]] ) + \[Beta][[2]]/\[Beta][[1]] -
datalv1[[n, 4]]);

```

a

附录四 问题二 2014 年等待时间的灰色预测 Mathematica 10 代码

```

Clear["Global`*"]
data14sj0 = {{10.3, 9.7, 9.4, 9.6, 9.466138561263733}, {10.1,
9.6, 9.9, 9.7, 9.832824917083101}, {6.1, 5.5, 6.3, 6.1,
6.570329020232336}, {10.8, 10.3, 10.8, 10.9,
11.274150514873}, {9.1, 8.7, 8.4, 8.2,
7.942261694028083}, {13.3, 12.8, 12., 12.4,
11.997782638595481}, {8.9, 8.5, 8.3, 8.6,
8.568051499343895}, {10.9, 10.6, 10.1, 9.8,
9.385125716665534}, {11.2, 10.8, 11., 11.1,

```

```

11.268942249272435`}, {9.1`, 8.8`, 8.7`, 8.4`,
8.241065521951555`}};
data14sj1 = {{10.3`, 20.`, 29.4`, 39.`, 48.46613856`}, {10.1`, 19.7`,
29.6`, 39.3`, 49.13282492`}, {6.1`, 11.6`, 17.9`, 24.`,
30.57032902`}, {10.8`, 21.1`, 31.9`, 42.8`, 54.07415051`}, {9.1`,
17.8`, 26.2`, 34.4`, 42.34226169`}, {13.3`, 26.1`, 38.1`, 50.5`,
62.49778264`}, {8.9`, 17.4`, 25.7`, 34.3`, 42.8680515`}, {10.9`,
21.5`, 31.6`, 41.4`, 50.78512572`}, {11.2`, 22.`, 33.`, 44.1`,
55.36894225`}, {9.1`, 17.9`, 26.6`, 35.`, 43.24106552`}};
a = Array[x, 10];
For[n = 1, n < 11, n++,

B1 = -0.5 (data14sj1[[n, 1]] + data14sj1[[n, 2]]);
B2 = -0.5 (data14sj1[[n, 2]] + data14sj1[[n, 3]]);
B3 = -0.5 (data14sj1[[n, 3]] + data14sj1[[n, 4]]);
B4 = -0.5 (data14sj1[[n, 4]] + data14sj1[[n, 5]]);
Y1 = data14sj0[[n, 2]];
Y2 = data14sj0[[n, 3]];
Y3 = data14sj0[[n, 4]];
Y4 = data14sj0[[n, 5]];
B = {{B1, 1}, {B2, 1}, {B3, 1}, {B4, 1}};
Y = {Y1, Y2, Y3, Y4};
\[Beta] = Inverse[(B\[Transpose].B)].B\[Transpose].Y;
x[n] = ((data14sj0[[n, 1]] - \[Beta][[2]]/\[Beta][[
1]]) E^(-5 \[Beta][[1]]) + \[Beta][[2]]/\[Beta][[1]] -
data14sj1[[n, 5]]);];
a

```

附录五 问题二 2014 年满载率的灰色预测 Mathematica 10 代码

```

Clear["Global`*"]
data14lv0 = {{64.`, 65.`, 68.`, 66.`, 67.32590941860872`}, {70.`,
76.`, 73.`, 75.`, 73.65988068093247`}, {56.`, 58.`, 60.`, 59.`,
59.997223916837356`}, {70.`, 75.`, 74.`, 72.`,
70.71425761738874`}, {63.`, 65.`, 67.`, 68.`,
69.70405901658887`}, {65.`, 67.`, 70.`, 68.`,
69.3261234681213`}, {67.`, 70.`, 72.`, 69.`,
69.34981841543959`}, {72.`, 75.`, 79.`, 80.`,
83.07457548908678`}, {76.`, 78.`, 75.`, 73.`,
70.43333009510025`}, {57.`, 59.`, 60.`, 61.`,
62.022174316647124`}};
data14lv1 = {{64.`, 129.`, 197.`, 263.`, 330.3259094`}, {70.`, 146.`,
219.`, 294.`, 367.6598807`}, {56.`, 114.`, 174.`, 233.`,
292.9972239`}, {70.`, 145.`, 219.`, 291.`, 361.7142576`}, {63.`,
128.`, 195.`, 263.`, 332.704059`}, {65.`, 132.`, 202.`, 270.`,

```

```

339.3261235`}, {67.` , 137.` , 209.` , 278.` , 347.3498184`}, {72.` ,
147.` , 226.` , 306.` , 389.0745755`}, {76.` , 154.` , 229.` , 302.` ,
372.4333301`}, {57.` , 116.` , 176.` , 237.` , 299.0221743`}}};
a = Array[x, 10];
For[n = 1, n < 11, n++,

B1 = -0.5 (data14lv1[[n, 1]] + data14lv1[[n, 2]]);
B2 = -0.5 (data14lv1[[n, 2]] + data14lv1[[n, 3]]);
B3 = -0.5 (data14lv1[[n, 3]] + data14lv1[[n, 4]]);
B4 = -0.5 (data14lv1[[n, 4]] + data14lv1[[n, 5]]);
Y1 = data14lv0[[n, 2]];
Y2 = data14lv0[[n, 3]];
Y3 = data14lv0[[n, 4]];
Y4 = data14lv0[[n, 5]];
B = {{B1, 1}, {B2, 1}, {B3, 1}, {B4, 1}};
Y = {Y1, Y2, Y3, Y4};
\[Beta] = Inverse[(B\[Transpose].B)].B\[Transpose].Y;
x[n] = ((data14lv0[[n, 1]] - \[Beta][[2]]^\[Beta][[
1]]) E^(-5 \[Beta][[1]] ) + \[Beta][[2]]^\[Beta][[1]] -
data14lv1[[n, 5]]);];
a

```

附录六 图二的 Mathematica 10 代码

```

ClearAll["Global`*"]
a = {4685, 1976, 1541, 1517, 1600, 4668, 5244, 6129, 8029, 8952, 6913,
10024, 16061, 17928, 14093, 9881, 10380, 12448, 11628, 10988,
11520, 10953, 7613, 6473, 4896, 2553, 1493, 992, 1088, 2945, 8560,
22997, 33041, 23649, 13972, 14204, 13993, 18240, 18708, 18568,
21269, 27992, 30196, 22160, 23388, 39717, 45024, 31573, 6036, 2512,
1296, 1312, 1124, 2497, 3845, 9505, 9309, 7556, 7212, 7973, 8613,
9828, 8696, 7700, 7417, 9477, 9061, 9328, 8592, 7188, 5541, 5392,
3017, 1192, 944, 825, 897, 1905, 3560, 7588, 9184, 6900, 5736,
5953, 5872, 7376, 6761, 6112, 5753, 8172, 9488, 8800, 8637, 11821,
8157, 4145, 2061, 1313, 893, 781, 988, 1652, 5132, 11100, 13669,
8640, 5884, 5569, 5053, 6308, 6173, 6017, 6108, 7904, 8485, 7649,
7340, 11261, 8701, 3637, 1933, 1004, 857, 696, 969, 1888, 4488,
10200, 13932, 9968, 5892, 5408, 5169, 6857, 6388, 6213, 6288, 8756,
8828, 7908, 7665, 11505, 9660, 3953, 2221, 1356, 965, 865, 840,
1361, 4449, 9273, 13201, 10236, 5220, 4928, 5192, 6240, 6053, 5940,
6552, 9620, 10692, 8829, 7516, 12592, 10128, 3821};
data1 = 1/a;
c = {58.64, 56.05, 60.31, 68.08, 62.96, 69.74, 63.33, 46.83, 53.01,
56.02, 53.6, 53.56, 61.05, 62.97, 54.09, 58.38, 57.49, 64.16,
57.32, 58.58, 65.0, 72.42, 60.18, 59.5, 25.98, 21.15, 21.48, 27.35,

```

```

26.31, 23.22, 26.61, 40.27, 46.67, 39.67, 26.51, 23.89, 25.43,
31.88, 31.04, 28.96, 27.17, 31.86, 35.93, 30.08, 30.28, 43.13,
46.47, 32.25, 72.17, 65.92, 60.88, 66.37, 63.37, 57.48, 62.23,
68.74, 58.19, 49.57, 51.85, 50.61, 56.22, 56.9, 52.67, 55.78,
50.13, 53.95, 61.95, 70.75, 53.33, 57.0, 49.64, 48.48, 60.62,
64.63, 64.07, 62.99, 62.29, 59.45, 64.24, 62.74, 63.99, 49.42,
37.96, 43.78, 46.78, 51.63, 46.66, 44.65, 47.23, 58.3, 62.02,
57.63, 55.88, 77.29, 71.69, 44.36, 50.46, 51.6, 50.68, 55.27, 49.9,
49.26, 61.32, 74.78, 73.27, 54.73, 39.24, 38.39, 42.09, 45.09,
47.44, 41.86, 42.86, 50.11, 61.21, 59.72, 48.87, 77.0, 72.66,
48.25, 43.3, 47.1, 58.18, 54.52, 63.95, 48.77, 58.8, 70.06, 65.22,
48.68, 40.59, 41.51, 42.88, 46.85, 45.79, 43.61, 44.02, 56.0,
63.41, 56.29, 50.45, 70.05, 69.96, 48.37, 53.61, 54.01, 61.3,
59.12, 53.45, 39.77, 57.17, 76.48, 65.31, 53.12, 40.66, 35.43,
44.27, 48.6, 44.56, 45.94, 40.48, 52.1, 63.93, 63.17, 48.82, 69.45,
74.9, 46.95};
data2 = 1/(c + 300);
Array[x, 168];
Array[p, 168];
For[k = 1, k < 169, k++,
  x[k] = (data1[[k]] - Min[data1])/(Max[data1] - Min[data1]);
For[s = 1, s < 169, s++,
  p[s] = (data2[[s]] - Min[data2])/(Max[data2] - Min[data2]);
quan2 = Array[x, 168];
quan1 = Array[p, 168];
Array[t, 168];
For[r = 1, r < 169, r++,
  t[r] = 0.7 (1 - E^-((quan1[[r]] - 0.5)^2/0.197703^2)) quan1[[r]] +
    0.3 (1 - E^-((quan2[[r]] - 0.5)^2/0.197703^2)) (1 - quan2[[r]]);
X = Array[t, 168];
Show[ListLinePlot[a, PlotRange -> All,
  Ticks -> {{{13, "九月 3 号"}, {37, "九月 4 号"}, {61, "九月 5 号"}, {85,
    "九月 6 号"}, {109, "九月 7 号"}, {133, "九月 8 号"}, {157, "九月 9 号"}},
  Automatic}, PlotStyle -> {Dashed, Thick}, Axes -> {True, False}],
ListLinePlot[20000 X,
  Ticks -> {{{13, "九月 3 号"}, {37, "九月 4 号"}, {61, "九月 5 号"}, {85,
    "九月 6 号"}, {109, "九月 7 号"}, {133, "九月 8 号"}, {157, "九月 9 号"}},
  Automatic}, PlotStyle -> {Red}, PlotRange -> All,
  Axes -> {True, False}],
Plot[0.3019 20000, {x, 0, 168}, PlotStyle -> {Black, Dashing[Tiny]]]

```

附录七 图三的 Mathematica 10 代码

```

Clear["Global`*"]
d = {4219., 4901., 5054., 3979., 4538., 4896., 5367., 5095.,

```

```

4314., 5159., 5267., 5310., 8132., 9913., 11272., 12971.,
16592., 16354., 17631., 19097., 20738., 24639., 25895.,
24669., 28157., 28101., 26605., 26122., 15849., 15636.,
16629., 16924., 19532., 22264., 25778., 33178., 33173.,
32479., 32450., 33486., 34967., 24228., 21497., 18768.,
16017., 14669., 15854., 84065., 124069., 119336., 108799.,
107787., 104307., 97752., 114127., 123967., 129511.,
145343., 122902., 148264., 133152., 142221., 146878.,
138827., 115327., 103941., 85837., 71709., 58666., 48488.,
47567., 40647., 41955., 40431., 36509., 32266., 31404.,
31922., 30595., 30601., 33381., 29161., 23705., 21423.,
18855., 17898., 22724., 24601., 19372., 17365., 16203.,
19764., 18633., 20446., 19299., 17733., 17841., 17117.,
16921., 16379., 18330., 17827., 17012., 14948., 14897.,
13990., 14379., 14535., 14339., 12976., 13639., 12092.,
12043., 12806., 15043., 14279., 13483., 11237., 11431.,
13989., 13707., 12121., 11674., 11608., 11079., 11073.,
10998., 10007., 11950., 11752., 12086., 9988., 9598.,
9403., 9082., 13250., 13991., 10251., 9126., 10563.,
10069., 8683., 9734., 10627., 9099., 7755., 7796., 8478.,
8553., 11185., 9376.};
a = Array[dd, 151];
For[k = 1, k < 152, k++, dd[k] = \!\(
\*UnderoverscriptBox[\(\[Sum]\), \(\text{i} = 1\), \(\text{k}\)]\(\text{d}[\text{d}[\text{i}]]\)\)\);
b = d/a;
Show[ListLinePlot[b, AxesLabel -> {"日期/天", "增长率"}],
Plot[c1/x /. {c1 -> 1.01570201116571}, {x, 1, 151},
PlotRange -> All, PlotStyle -> {Dashed, Red}]]

```

附录八 图四和图五的 Mathematica 10 代码

```

p = Predict[{0 -> 0, 10 -> 0.926499, 16 -> 0.967951, 10 -> 0.81863,
13 -> 1.02049}, Method -> "NeuralNetwork"];

Plot[p[x], {x, 0, 20}, AxesLabel -> {"补贴额度/元", "收益指数"}]

Plot[p[x]/x, {x, 0, 20}, AxesLabel -> {"补贴额度/元", "性价比"}]

```