

高速公路车辆通行压力及最佳配置研究

摘要

高铁的修建一方面能够缓解普通列车的运行压力，另一方面对高速公路的运行也起到了减压的作用。研究高铁的数量配置是有实用价值的。

针对问题一：分析高速公路车辆通行压力情况，建立起二级评价指标体系，分为4个一级指标，包括道路条件、交通条件、交通管制条件、其他条件，以及10个二级指标。然后建立起AHP-模糊综合评价模型，用层次分析法确定各级指标权重，并使用Matlab软件进行求解；使用模糊综合评价法对高速公路车辆通行压力进行评价，将高速公路车辆通行压力划分为5个等级，并设置不同等级的得分范围，以哈尔滨市为例，得到其高速公路车辆通行压力水平得分为6.3，等级为良好，并对模型的可推广性及广泛性进行了说明。接着分析高铁运营对高速公路车辆通行压力产生影响的具体指标，收集数据，将高铁运营后高速公路车辆通行压力水平的得分求解出来，为8.1，较之前提高了1.8，等级提升为优秀。最后使用Excel软件对高铁运营前后高速公路车辆通行压力水平的两组得分进行显著性分析，得到P值为0.018725，大于0.01，小于0.05，说明两组数据具有显著差异，即高铁的运营对于高速公路的车辆通行压力具有显著的缓解作用。

针对问题二：选取不同发展水平的五个城市：一线城市北京、新一线城市成都、二线城市济南、三线城市绵阳和四线城市乐山。引入消费水平系数M，M的值越大，出行者越倾向于选择路费高的最短路；M的值越小，出行者越倾向于选择路费低的次短路或其他路费更低的道路。建立目标规划模型，以最佳的高铁配置数量为目标，并将其转化为高铁对高速公路车流量分流能力L最大，并使用蚁群算法对规划模型衍生出的最佳路径问题进行求解，使用C语言进行编程，得到最终结果，即各城市的最佳高铁配置数量，具体结果见表16。

在文章的最后对模型的优缺点进行评价，并给出模型的改进意见。

关键词：指标体系 AHP-模糊综合评价模型 目标规划 蚁群算法

一、 问题重述

随着人们生活质量的不断提高，几乎每家都拥有了私家车，这也就造成了每当节假日来临，高速公路就会出现拥堵情况的原因之一。但是随着中国现代“新四大发明”之一的高铁迅猛发展，给人们的出行带来了极大的便利，因此远途出行的人们也逐渐将高铁作为自己出行的首选交通工具。高铁的修建一方面能够缓解普通列车的运行压力，另一方面对高速公路的运行也起到了减压的作用。

我们需要完成以下两个问题：

(1) 高铁的开通，一部分人们便会选择高铁出行，从而会使得高速公路的车辆有所减少，请你选取合适的指标，分析高铁的开通对该高速公路的车辆通行压力是否有所减缓，并分析是否显著。

(2) 高铁既快捷，又舒适，但是相对于普通列车出行价格相对昂贵，因此不同地域的人出行方式的选择将会有所差异，请你选择发展不同的城市，尝试给出你所选城市高铁配置的最佳数量。

二、 问题分析

这是一个研究高铁开通的影响及城市高铁配置的问题，首先需要列出合理的指标体系，对各指标进行量化描述，结合指标体系建立有关高速公路车辆通行压力的评价模型，分析高铁的开通对其是否具有减缓作用，并进行显著性分析；然后选择发展水平不同的城市，分析不同城市的人口出行方式，并给出各个城市高铁配置的最佳数量。

2.1 道路通行能力

道路通行能力是道路能够疏导或处理交通流的能力。我国定义为，道路通行能力是指道路上某一车道或某一断面处，单位时间内可能通过的最大交通量车辆或行人数，用辆或用辆昼夜表示，车辆多指小汽车，当有其它车辆混入时，均采用等效通行能力的当量小客车为单位。通行能力是指所分析的道路、设施没有任何变化，并假定其具有良好的气候条件和路面条件下的通过能力，如条件有任何变化都会引起通行能力的变化。

道路通行能力按车辆运行状态的特征可分为：

1) 路段通行能力：

城市道路路段可能通行能力考虑道路和交通条件的影响，并对基本通行能力进行修正后得到的通行能力，实际上指的是道路所能承担的最大交通量。

2) 交叉口的通行能力：

是指各相交道路进口处通行能力之和（以进口处车道的停车线作为基准面，凡是通过该断面的车辆被认为已通过交叉口），而每个进口处通行能力分为直行、右转、左转三种情况。

3) 合流、分流状态下的通行能力:

支路与主路之间合流、分流时可以承受的最大交通车流量改变程度。

4) 交织运行状态的通行能力:

是指两股或两股以上交通运行方向基本相同的车流,先实现合流而后分流的整个运行过程所需的路段。交织区的通行能力和运行速度同交织区长度、车道数、交织流量比、总交通量及交织区车道构造等因素有关。

2.2 车辆通行压力

通行能力实质上是道路负荷性能的一种度量,它既反映了道路疏通交通的最大能力,也反映了在规定特性前提下,道路所能承担车辆运行的极限值。

当实际的车辆通行情况超过道路通行能力时,就会造成车辆通行压力。

三、 模型假设

1. 忽略本文所考虑指标以外的其他因素对高速公路车辆通行能力水平的影响;
2. 研究过程中选择的城市具有一定代表性;
3. 所采用的数据准确,能代表该地区水平。

四、 符号说明

由于每个模型所用符号很多,因此在模型具体建立过程中再列出所用符号,不在此做具体说明。

五、 问题一的模型建立与求解

5.1 问题一的分析

高铁的开通,一部分人们便会选择高铁出行,从而会使得高速公路的车辆有所减少。我们需要列出合理的指标体系,对各指标进行量化描述,结合指标体系建立有关高速公路车辆通行压力的评价模型,分析高铁的开通对其是否具有减缓作用,并进行显著性分析。

5.2 高速公路车辆通行压力指标体系的建立及说明

一、 道路条件:

1.基本路段通行能力:

基本路段通行能力是指道路与交通处于理想情况下,每一条车道(或每一条道路)在单位时间内能够通过的最大交通量^[1]。理想的道路条件主要是车辆组成单一的标准型

汽车，在一条车道上以相同的速度，连续不断的行驶，各车辆之间保持与车速相适应的最小车头间隔，且无任何方向的干扰；车道宽度应不小于 3.65m。(我国公路规定为 3.75m)，路旁的侧向余宽不小于 1.75m，纵坡平缓，并有开阔的视野、良好的平面线形和路面状况。定义公式为：

$$\text{道路通行能力} = \frac{3600}{t_0} = \frac{3600}{l_0 / (v/3.6)} = \frac{1000v}{l_0} \text{ (辆/h)} \quad (1)$$

式中：(s)--行车速度(km/h)； t -驾驶员反映时间(s)；

$$\text{其中：} l_0 = l_{\text{反}} + l_{\text{制}} + l_{\text{安}} + l_{\text{车}} = \frac{V}{3.6}t + \frac{V^2}{254\phi} + l_{\text{安}} + l_{\text{车}}$$

l_0 —车头最小间距(m)； t —驾驶员反映时间(s)，可取 1.2 s 左右； $l_{\text{反}}$ —驾驶员在反应时间内车辆驾驶的距离(m)； $l_{\text{制}}$ —车辆的制动距离(m)； $l_{\text{安}}$ —车辆间的安全距离(m)，可取 5m； $l_{\text{车}}$ —车辆平均长度(m)，小汽车可采用 5m，载货汽车为 7m，大型客车为 9m。

2.道路网密度：

道路网密度是指建成区内道路长度与建成区面积的比值（道路指有铺装的宽度 3.5 米以上的路）用 km/km^2 表示^[2]。是对道路长度与用地面积间发展关系的限定，它从长度上描述了不同城市规模应有的道路发展水平。

(1) 计算公式为：

$$\text{道路网密度} = \frac{L}{S} \quad (2)$$

其中： L —道路长度； S —全区总面积。

(2) 表 1 为道路网密度分级标准：

表 1：道路网密度分级表

评价标准等级	一	二	三	四	五
P18	[7.0, 9.0]	[6.0, 7.0)	[5.0, 6.0)	[4.0, 5.0)	[1.0, 4.0)
指数	[90, 100]	[80, 90)	[70, 80)	[60, 70)	[0, 60)

二、 交通条件：

1.路段服务水平——饱和度：

饱和度也可称为负荷系数，为服务交通量与实际通行能力的比值，它反映了道路的实际负荷情况。对于以集散交通为主的次干路、支路，以通行能力为主，饱和度接近 1，甚至饱和度大于 1，只是行车缓慢，赢得了较大的通行能力和行车安全。行车速度有所降低，服务水平降至 E 级，交通环境处于“通而不畅”的交通状态，与道路的功能一致^[2]。

(1) 定义式如下：

$$\text{饱和度} = \frac{\text{实际流量}}{\text{通行能力}} \quad (3)$$

交通量与可能通行能力之间的关系是随着道路交通流量的增加，饱和度逐渐增长，行车速度呈下降趋势，接近时，行车速度降低到道路允许速度，一般为道路设计计算行车速度的倍数。

(2) 利用饱和度衡量服务水平的分级标准：

由于饱和度不好计算，将其与平均运行速度建立起对应关系，如下表：

表 2：我国大中城市道路服务水平划分标准(本表参考《城市交通管理指南》^[3])

等级	饱和度	运行特征	平均运行速度 (km/h)
A	≤0.35	自由运行交通流(畅通)	44~48
B	0.35~0.55	合理的自由交通流(稍大延误)	40~44
C	0.55~0.75	稳定交通流(能接受的延误)	36~40
D	0.75~0.90	接近不稳定交通流(能忍受的延误)	32~36
E	0.90~1.00	极不稳定交通流(拥挤、不能忍受的延误)	28~32
F	>1.0	强制性车流阻塞(拥挤、不能忍受的延误)	24~28

2.通行能力折减系数：

在一些城市主干道上，同一行使方向的车道数往往不止一条，在多车道的情况下，同向行驶的车辆由于超车、超越、停车等原因影响另一车道的通行能力。一般越靠近路中心线的车道，其影响越小^[4]。因此，在无分隔带的同向车道上，靠近路中心线的车道通行能力最大，靠近缘石的车道通行能力最小。系数分级标准如下：

表 3：多车道折减系数分级标准

车道条数	1	2	3	4	5
通行能力折减系数	1.00	0.80~0.89	0.65~0.78	0.50~0.65	0.40~0.52

● **交叉口通行能力折减系数 $a_{交}$ ：**

城市道路的一个特点就是纵横交叉的街道形成的许多交叉口，交叉口是影响城市道路通行能力的主要因素，尤其是交叉口的间距较小时，它对通行能力往往起控制作用。

(1) 定义式如下^[5]：

$$a_{交} = \frac{\text{交叉口之间无阻的行程时间}(s)}{\text{交叉口之间实际的行程时间}(s)} \quad (4)$$

城市道路的一个特点就是纵横交叉的街道形成的许多交叉口，交叉口是影响城市道路通行能力的主要因素，尤其是交叉口的间距较小时，它对通行能力往往起控制作用。

(2) 计算公式如下：

$$\alpha_{交} = \frac{\frac{l}{v'}}{\frac{l}{v'} + \beta_1 \frac{v'}{2a} + \beta_2 \frac{v'}{2b} + \Delta} \quad (5)$$

式中： l —交叉口之间距离(m)； v' —路段上的行车速度(m/s)， a —启动时的平均加速度(m/s²)。据观测资料：小型汽车 $a = 0.60 \sim 0.70$ ，中型货车 $a = 0.49 \sim 0.53$ ，大型货车及大型客车 $a = 0.42 \sim 0.46$ ，交接公共车辆 $a = 0.43 \sim 0.49$ ，以中型货车为主的各种车型混合行驶的平均值 $a = 0.50$ ； b —制动时的平均减速度(m/s²)，据观测资料：小型汽车 $b = 1.66$ ，大型汽车 $b = 1.30$ ，各种车型混合行驶的平均值 $b = 1.50$ ； β_1 和 β_2 ——数值取决于 v'_A/v' 的比值，其中： $\beta_1 = (1 - v'_A/v')^2$ ， $\beta_2 = (1 - v'_B/v')^2$ ； v'_A 和 v'_B ——车辆到达交叉口的车速(m/s)； Δ ——车辆在交叉口上的停候时间(s)，当 $v'_A(v'_B) = 0$ 时，则 Δ 为红灯的一半，一般 $\Delta = 15 \sim 20s$ ，当 $v'_A(v'_B) \neq 0$ 时，则 $\Delta = 0$ 。 β_1 及 β_2 的值如下表所示：

表 4: β_1 及 β_2 的值

$v'_A/v', v'_B/v'$	0	0.50	0.55	0.60	0.65	0.7	0.75	0.85
β_1, β_2	1	0.25	0.20	0.16	0.12	0.09	0.06	0.04

● **车道宽度折减系数 $a_{\text{车道}}$ ：**

道路通行能力与车道宽度有密切关系，当车道的宽小于必需的 3.5m 时，必然会影响到车辆的前进速度，而车速的降低则意味着通行能力的减小。因此，用作为道路宽度不足 3.50 时通行能力的折减系数。系数分级标准^[6]如下：

表 5: 车道宽度折减系数分级标准

车道宽度 (m)	3.50	3.25	3.00	2.75
$a_{\text{车道}}$	1.00	0.94	0.85	0.77

三、 交通管制条件：

以各交叉路口的饱和度表示各入口引道的服务水平，其服务水平受到交通控制，通过交叉口时间、延误时间、停车次数和频率等影响。处在交叉口位置时，会产生直行、右转、左转、直右混合、直左混合 5 种情况，影响因素对其影响各不相同，综合各个指标构成以下指标。

1. 交叉口管制——行车时间损失：

指车辆在实际交通流条件下由于该车本身的加速、减速或停车而引起的时间延误，即与外部干扰无关的延误。

2. 延误时间：

是指车辆在行驶中，由于受到驾驶员无法控制的或意外的其他车辆的干扰或交通控制设施等的阻碍所损失的时间。如由于交通信号指示、车辆转弯和路旁停车等影响所损失的时间都属于交通延误。它是评价车流密度和道路设施的重要指标之一。表 6 为延误时间等级标准划分^[7]：

表 6：平均延误分级及其所反映的服务水平

等级	平均延误时间(s)	运行特征
A	≤ 0.35	运行延误很小，车流畅通
B	0.35~0.55	运行延误稍大，行驶空间受阻
C	0.55~0.75	停车数量增多，出现轻微阻塞
D	0.75~0.90	影响较为严重，延误进一步增大，大多数车须停车
E	0.90~1.00	延误达到可接受的极限
F	>1.0	延误不能接受，交叉口严重受阻处于超饱和状态

3. 行车方向改变概率 η ：

包括直行车所占百分率、右转车所占百分率、左转车所占百分率三部分，分别为该行驶方向的车辆数占整体车流量的百分比，反映车辆在交叉口处做出的路线决策。

4. 交叉口道路基本通行能力

- 一条直行车道通行能力：计算公式为：

$$N_{直} = \frac{3600}{T_{周}} \times \frac{t_{绿} - t_{损}}{t_{间}} \quad (6)$$

式中： $T_{周}$ —信号灯周期时间，一般取 60~90s； $t_{绿}$ —每个信号灯周期内的绿灯时间； $t_{损}$ —一个手内绿灯损失的时间，一般只计车辆加速时间损失，不计反应和启动的时间损失（绿灯前的黄灯时间已准备好），加速损失时间 $t_{加} = \frac{v}{2a}$ （ v ：直行车辆通过交叉口的车速，一般取 15km/h； a ：平均加速度，小汽车 $a = 0.6 \sim 0.7m/s^2$ ，中型卡车 $a = 0.5 \sim 0.6m/s^2$ ，中型车 $a = 0.4 \sim 0.5m/s^2$ ）； t_{jian} —前后车连续通过停车线的平均时间间隔，小汽车车流取 2.5s，大型车取 2.5s。

- 交通管制条件——右转车道通行能力

- (1) 一条专用右转车道通行能力：

$$N_{右} = \frac{3600}{t_{右}} \quad (\text{辆/h}) \quad (7)$$

式中： $t_{右}$ —前后两右转车连续驶过停车线的时间间隔，小汽车 $t_{右} = 3.0 \sim 3.6s$ ，平均 4.5s。一条直右混行专用车道通行能力：

- (2) 一条直右混行专用车道通行能力：

$$N_{直右} = N_{直} \left(1 - \frac{n_{右}}{2}\right) K \quad (\text{辆/h}) \quad (8)$$

式中： $n_{右}$ 为右转车所占百分率； K 为直、右车辆混行时，因相互干扰的折减系数，可取 1.5。

● 交通管制条件——左转车道通行能力

(1) 一条专用左转车道通行能力：

$$N_{左} = n \times \frac{3600}{T_{周}} \quad (\text{辆/h}) \quad \left(n = \frac{t_{黄lv} - \frac{v_{左}}{2}}{t_{左}} \right) \quad (9)$$

式中： n ——一个周期内允许左转的车辆数， $t_{黄lv}$ ——一个周期内在专用于左转车黄绿灯的时间(s)； $v_{左}$ ——左转弯车辆速度(t/s)； a ——左转弯车辆加速度(t/s²)， $t_{左}$ ——左转弯通过停车线的(s)

(2) 一条直左混行专用车道通行能力：

$$N_{直左} = N_{直} \left(1 - \frac{3}{4} n_{左} \right) K \quad (10)$$

式中： $N_{左}$ 为左转弯车所占百分率，折减系数 K 可取 1.75。

四、 其他条件：

其它条件有气候、温度、地形、风力、心理等因素。但其中直接影响通行能力数值的主要因素有：车行道宽度及侧向净空、车行道数量、交通组成、驾驶员特性、道路纵坡、横向干扰与视距等。

5.5 评价指标体系建立

根据以上对各指标的量化分析，建立起关于高速公路车辆通行压力指标体系，如下表所示：

表 7：高速公路车辆通行压力指标体系

目标	一级指标	二级指标
道路通行能力	道路条件 B1	基本路段通行能力 C1
		道路网密度 C2
	交通条件 B2	服务水平：饱和度 C3
		通行能力折减系数 C4
	交通管制条件 B3	行车时间损失 C5
		延误时间 C6
		行车方向改变概率 C7
		交叉口道路基本通行能力 C8
	其他条件 B4	气候、地理条件 C9
		驾驶员特性 C10

5.3 AHP—模糊综合评价模型原理及过程

高速公路车辆通行压力的评价，从本质上来说就是定性评价与定量评价的结合，定性评价是不可或缺的。AHP—模糊综合评价法可以有效地将高速公路车辆通行压力水平中的定性评价进行定量化处理，基本上实现总体上的量化评价，提高两者评价的可靠度。

本文试把层次分析法(AHP)应用于高速公路车辆通行压力评价体系中的评价指标权重的确定，同时，借用模糊数学的模糊综合评价的方法，将各位评价委员的评价结果进行模糊综合处理，科学地界定地区的高速公路车辆通行压力水平。

AHP—模糊综合评价模型主要由两个部分组成，第一部分：层次分析法，第二部分：模糊综合评价。其中，模糊综合评价是在层次分析法的基础上进行的，两者相辅相成，共同提高了评价的可靠性与有效性^[11]。

5.3.1 层次分析法——科学地确定评价指标的权重

层次分析法是从定性分析到定量分析综合集成的一种典型的系统工程方法，它将人们对复杂系统的思维过程数学化，将人的主观判断为主的定性分析进行定量化，将各种判断要素之间的差异数值化，帮助人们保持思维过程的一致性，适用于复杂的模糊综合评价系统，是目前一种被广泛应用的确权重的方法^[11]。

运用层次分析法确定高速公路车辆通行压力评价指标的权重，可以按以下的步骤进行：

Step 1: 建立高速公路车辆通行压力评价体系的递阶层次结构

即对所评判的 N 个高速公路车辆通行压力指标进行层次分析，确立清晰的分级指标体系，给出由高速公路车辆通行压力构成的的指标集和子指标集，分别表示为：高速公路车辆通行压力总指标集 $V = \{V_1, V_2, \dots, V_N\}$ ，子指标集 $V_i = \{V_{i1}, V_{i2}, \dots, V_{ik}\}$ ， V_i 代表高速公路车辆通行压力指标， N 为高速公路车辆通行压力指标个数，上述指标集为按高速公路车辆通行压力指标编号进行的简单排序。

Step 2: 确定两两比较判断矩阵^[8]

查阅文献，利用 1-9 比例标度法分别对每一层次的评价指标的相对重要性进行定性描述，并用准确的数字进行量化表示，进而确定判断矩阵。

表 8: 判断矩阵及标度的含义表

标度	含义
1	表示表示两个指标相比，具有同样重要性
3	表示两个指标相比，一个指标比另一个指标稍微重要
5	表示两个指标相比，一个指标比另一个指标明显重要
7	表示两个指标相比，一个指标比另一个指标强烈重要
9	表示两个指标相比，一个指标比另一个指标极端重要

2, 4, 6, 8 为上述相邻判断的中值, 若指标 a 和 b 比较得 a_{ij} , 则指标 b 和 a 比较的判断 $1/a_{ij}$ 。高速公路车辆通行压力总指标集 $V = \{V_1, V_2, \dots, V_N\}$ 中各高速公路车辆通行压力指标相对于总的评价目标, 两两重要性比较的判断矩阵如下:

$$A = \begin{bmatrix} 1 & V_{12} & \cdots & V_{1N} \\ V_{21} & 1 & \cdots & V_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ V_{N1} & V_{N2} & \cdots & 1 \end{bmatrix} = (V_{ij})_{N \times N} \quad (V_{ij} = \frac{1}{V_{ji}})$$

其中, V_{ij} 表示对于总的评价目标而言, 元素 V_i 对 V_j 相对重要性的判断值, 该矩阵的特点是对角线上的元素为 1, 即每个指标相对于自身的重要性为 1。

子指标集 $V_i = \{V_{i1}, V_{i2}, \dots, V_{ik}\}$ 中各指标相对于指标 V_i 而言, 两两重要性比较的判断矩阵如下, 其中, f_{ij}^i 表示相对于 V_i 一级评价指标而言, 指标 V_{il} 对 V_{ij} 相对重要性的判断值。

$$B_i = \begin{cases} v_{i1} \\ v_{i2} \\ \cdots \\ v_{ik} \end{cases} \begin{bmatrix} 1 & f_{12}^i & \cdots & f_{1k}^i \\ f_{21}^i & 1 & \cdots & f_{2k}^i \\ \cdots & \cdots & f_{12}^i & \cdots \\ f_{k1}^i & f_{k2}^i & \cdots & 1 \end{bmatrix} = f_{k \times k}^i \quad (i = 1, 2, \dots, N) \quad (f_{ij}^i = \frac{1}{f_{ji}^i})$$

Step 3: 运用和积法求解判断矩阵, 分别得出在单一准则 M 和 V_i 下被比较高速公路车辆通行压力元素的相对权重——即层次单排序^[9]

- 1) 将矩阵 A 中各元素按列做归一化处理;
- 2) 将处理后得到的矩阵按行分别相加;
- 3) 将所得的行向量分别做归一化处理, 得到单一准则下所求各被比较元素的权重向量 ω ;

- 4) 按下列公式计算唯一最大特征根 λ_{\max} , $\lambda_{\max} = \sum_{i=1}^n \frac{(A\omega)_i}{n\omega_i}$ (其他的判断矩阵 B_i 求法同上)。

同上)。

Step 4: 层次总排序——计算同层次高速公路车辆通行压力指标组合权重的矩阵:

(H_1, H_2, \dots, H_m)

Step 5: 一致性检验^[10]

- 1) 首先计算一致性指标 CI , $CI = \lambda_{\max} - n / n - 1$, 其中 n 为判断矩阵的阶数;
- 2) 平均随机一致性指标 RI ;
- 3) 计算一致性比例 CR , 其中 $CR = CI / RI$ 。当 $CR < 0.1$ 时, 一般认为判断矩阵的一致性是可以接受的, 其中, CR 的值越小, 说明判断矩阵偏离实际情况的值越小, 就越接近于现实情况。

因此, 从以上我们可以看出, 运用层次分析法求解各个评价指标的权重时, 只需评价人员给出各个评价指标的两两相对重要性的一个定性的描述, 然后通过 AHP 方法就可

以比较精确的求出各个评价元素的权重，层次分析法将定性的描述与定量的计算很好地结合了起来，这些都是以严格的科学理论作为基础的，这就大大加强了评价过程中的科学性和有效性。

5.3.2 模糊综合评价

论域 U 中的模糊集合 A 是以隶属函数 μ_A 为表征的集合，即 $\mu_A:U \rightarrow [0,1]$ ，对任意 $u \in \mu_A(u)$ ， $\mu_A(u) \in [0,1]$ ，称 $\mu_A(u)$ 为元素 u 对于 A 的隶属度，它表示 u 属于 A 的程度。 $\mu_A(u)$ 的值越接近于1，表示元素 u 属于 A 的程度越高，作为上限，当 $\mu_A(u)=1$ 时，表示 u 完全属于 A ，反之， $\mu_A(u)$ 的值越接近于0，表示 u 属于 A 的程度越低，当 $\mu_A(u)=0$ 时，表示 u 完全不属于 A [12]。

模糊子集完全由它的隶属函数来刻画，隶属函数是模糊数学的最基本概念，借助隶属函数可以对模糊集合进行量化，从而利用精确的数学方法去分析和处理模糊信息。利用模糊综合评价可以有效的处理人们在评价过程中本身所带有的主观性，以及客观所遇到的模糊性现象 [11]。模糊综合评价通常按以下的步骤进行：

Step1: 建立对高速公路车辆通行压力作绩效评价结果的评判集 $U = \{u_1, u_2, \dots, u_n\}$ ，将评判集设定为： $U = \{great, good, common, pass, flunk\}$

Step2: 用隶属度分别描述各子指标相对于评判集 U 的隶属程度。

本论文中的隶属度即表示该高速公路车辆通行压力对于评判集的符合程度，得出单个地区高速公路车辆通行压力的模糊评判矩阵：

$$D_i = \begin{cases} v_{i1} \begin{bmatrix} S_{11}^i & S_{12}^i & \dots & S_{1n}^i \\ S_{21}^i & S_{22}^i & \dots & S_{2n}^i \\ \dots & \dots & S_{11}^i & \dots & \dots \\ v_{ik} \begin{bmatrix} S_{k1}^i & S_{k2}^i & \dots & S_{kn}^i \end{bmatrix} \end{bmatrix} \end{cases} \quad (i = 1, 2, \dots, m)$$

其中， S_{ij}^i 表示第 i 个一级评价指标下的第 l 个二级指标隶属于第 j 个评判等级的程度， i 为一级指标的数目， k 为第 i 个一级指标下的二级指标的数目， n 为评判集中评语的数目， S_{ij}^i 的意义及求法如下：

意义为：评判委员认为该高速公路车辆通行压力的该子指标方面的优秀程度比例。

首先由各位评价委员对每个高速公路车辆通行压力的被评价的子指标进行评定，然后通过统计整理的方法得到相对于子指标 V_{ij} 的若干个评语：其中包括 V_{ij1} 个 u_1 级评语， V_{ij2} 个 u_2 级评语，……，以及 V_{ijn} 个 u_n 级评语，则子指标层指标 V_{ij} 隶属于第 u_k 级评语的程度即隶属度 [12] 为：

$$S_{ij}^i = \frac{v_{ijk}}{\sum_{k=1}^n v_{ijk}} = v_{ijk} / (v_{ij1} + v_{ij2} + \dots + v_{ijn})$$

则子指标层指标 V_{ij} 的隶属度向量为: $S_j^i = (s_{11}^i, s_{12}^i, \dots, s_{in}^i)$, 由此就可得到 D_i 。

Step3: 一级模糊综合评判——利用模糊算子确定模糊关系矩阵 $R = (R_1, R_2, \dots, R_n)^T$

$$\text{其中, } R_i = (w_1^i, w_2^i, \dots, w_k^i) \begin{bmatrix} S_{11}^i & S_{12}^i & \cdots & S_{1n}^i \\ S_{21}^i & S_{22}^i & \cdots & S_{2n}^i \\ \cdots & S_{11}^i & \cdots & \cdots \\ S_{k1}^i & S_{k2}^i & \cdots & S_{kn}^i \end{bmatrix} = (r_{i1}, r_{i2}, \dots, r_{in})$$

$(w_1^i, w_2^i, \dots, w_k^i)$ 为第 i 个一级指标所属的二级指标的排序权重向量, 即各个指标的分配权重。

1) 二级模糊综合评判——确定被评价对象的最终评价结果

$$E = H \circ R = (H_1, H_2, \dots, H_m) \circ \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mn} \end{bmatrix} = (e_1, e_2, \dots, e_k, \dots, e_n)$$

其中, (H_1, H_2, \dots, H_m) 为在总目标下所有一级指标的排序权重向量, 即第一部分利用层次分析法所得出的权重向量。

2) 根据最大隶属度原则^[13], 确定被评价对象所属评判等级

若 $e_k = \max(e_1, e_2, \dots, e_k, \dots, e_n)$, 即 e_k 为该向量的第 K 个分量, 则根据模糊数学的最大隶属度原则, 被评价对象的评价结果属于第 K 等级。

3) 若由 P 类人员担任评价主体, 他们的综合评价结果分别为向量 E_1, E_2, \dots, E_p , 各类评价主体所占的权重分别为 T_1, T_2, \dots, T_k 。则总的评价结果为

$$E' = (T_1, T_2, \dots, T_k)(E_1, E_2, \dots, E_p)^T$$

得出的总的评价结果即为该地区高速公路车辆通行压力的综合评价分数。

5.4 评价模型求解——以哈尔滨市为例

由于数据收集问题, 在此仅以哈尔滨市为例, 进行模型求解。本模型可推广至全国范围及各城市及地区。

● 层次分析法求解指标权重

使用 Matlab 软件编程求解各级指标所占权重, 并将其整理在下表:

表 9: 指标权重表

指标层次	B1	B2	B3	B4
	0.3568	0.2672	0.2332	0.1428
C1	0.4988			
C2	0.5012			
C3		0.5012		
C4		0.4988		

C5			0.3518	
C6			0.2216	
C7			0.2931	
C8			0.1335	
C9				0.6463
C10				0.3537

● 模糊综合评价法评分

根据以上建立的模型，选取哈尔滨市为例，处理大量数据，并按照步骤求解：

(1)取因素集 $U = (p_1, p_2, \dots, p_n)$ ，将以上一级指标按顺序排列；

(2)取评语集 $R = \{\text{极差, 差, 一般, 良好, 优秀}\}$ ，R 为极差时，车辆通行压力最大；R 为优秀时，车辆通行压力最小；

(3)确定各因素的权重矩阵为： $A = [0.3568, 0.2672, 0.2332, 0.1428]$ ；

(4)确定模糊综合评判向量，对每个因素 x_i 做出评价；根据表 8，得出 10 个指标的判断矩阵，在此只展示前四个分别为：

$$\begin{aligned}
 x_1 &= [0.5, 0.5, 1.0, 0.7, 0.5, 0.5]; \\
 x_2 &= [0.7, 0.7, 1.0, 0.7, 0.6, 0.5]; \\
 x_3 &= [0.6, 0.5, 1.0, 1.0, 0.6, 0.5]; \\
 x_4 &= [0.6, 1.0, 1.0, 1.0, 0.6, 0.5];
 \end{aligned} \tag{11}$$

(5)模糊综合评判。得出 10 个指标的判断矩阵，进行向量合成运算，在此只展示前四个：

$$\begin{aligned}
 R_1 &= A \cdot x_1 = A \cdot [0.5, 0.5, 1.0, 0.7, 0.5, 0.5]^T = 0.62 \\
 R_2 &= A \cdot x_2 = A \cdot [0.7, 0.7, 1.0, 0.7, 0.6, 0.5]^T = 0.61 \\
 R_3 &= A \cdot x_3 = A \cdot [0.6, 0.5, 1.0, 1.0, 0.6, 0.5]^T = 0.69 \\
 R_4 &= A \cdot x_4 = A \cdot [0.6, 1.0, 1.0, 1.0, 0.6, 0.5]^T = 0.68
 \end{aligned} \tag{12}$$

● 求解结果：

哈尔滨市的 R 值水平在 0.63，高速公路车辆通行压力水平为良好，仍存在提升空间。

● 模型可推广性及广泛性

本问题在求解中，由于数据收集问题，选取哈尔滨市为例更便于展示模型结果。

但由于所建立的指标体系完善，具有可推广性。本模型在数据收集完整的情况下，可广泛应用于全国范围或任意规模的城市。

5.4 高铁对于高速公路车辆通行压力的缓解作用——造成影响的指标分析

我们对高铁运营后，高速公路车辆通行压力指标体系中各指标相应变化做出分析，

整理在下表中。

表 10: 高铁运营造成影响的指标（蓝色为提高的指标，红色为下降的指标）

目标	一级指标	二级指标	高铁运营后指标变化
道路 通行 能力	道路条件 B1	基本路段通行能力 C1	-----
		道路网密度 C2	-----
	交通条件 B2	服务水平：饱和度 C3	↓
		通行能力折减系数 C4	↑
	交通管制条件 B3	行车时间损失 C5	↓
		延误时间 C6	↓
		行车方向改变概率 C7	↓
		交叉口道路基本通行能力 C8	↑
	其他条件 B4	气候、地理条件 C9	-----
		驾驶员特性 C10	-----

由上表可知，高铁运营后，饱和度下降，通行能力折减系数提高，行车时间损失减少，延误时间减少，行车方向改变概率下降，交叉口道路基本通行能力提高，其他四个指标没有影响。

重新应用 AHP-模糊综合评价模型，对高铁运营后的高速公路车辆通行压力的 R 值进行计算（仍以哈尔滨市为例）：

(1)取因素集 $U = (p_1, p_2, \dots, p_n)$ ，将以上一级指标按顺序排列；

(2)取评语集 $R = \{\text{极差, 差, 一般, 良好, 优秀}\}$ ；

(3)确定各因素的权重矩阵为： $A = [0.3568, 0.2672, 0.2332, 0.1428]$ ；

(4)确定模糊综合评判向量，对每个因素 x_i 做出评价；根据表 8，得出 10 个指标的判断矩阵，在此只展示前四个分别为：

$$\begin{aligned}
 x_1 &= [0.5, 0.9, 1.0, 0.8, 0.5, 0.5]; \\
 x_2 &= [0.7, 1.0, 1.0, 0.9, 0.6, 0.5]; \\
 x_3 &= [0.1, 0.9, 1.0, 1.0, 0.2, 0.1]; \\
 x_4 &= [0.9, 1.0, 1.0, 1.0, 0.6, 0.5]; \tag{11}
 \end{aligned}$$

(5)模糊综合评判。得出 10 个指标的判断矩阵，进行向量合成运算，在此只展示前四个：

$$\begin{aligned}
 R_1 &= A \cdot x_1 = A \cdot [0.5, 0.9, 1.0, 0.8, 0.5, 0.5]^T = 0.72 \\
 R_2 &= A \cdot x_2 = A \cdot [0.7, 1.0, 1.0, 0.9, 0.6, 0.5]^T = 0.76 \\
 R_3 &= A \cdot x_3 = A \cdot [0.1, 0.9, 1.0, 1.0, 0.2, 0.1]^T = 0.81
 \end{aligned}$$

$$R_4 = A \cdot x_1 = A \cdot [0.9, 1.0, 1.0, 1.0, 0.6, 0.5]^T = 0.85 \quad (12)$$

● 由评价结果可知：

哈尔滨市的 R 值水平在 0.81，高速公路车辆通行压力水平为优秀，较之前提高了 0.18。由此可见，高铁的开通对于高速公路车辆通行压力有所减缓。

5.5 显著性分析

根据以上建立的 AHP-模糊综合评价模型，我们可以通过分析高铁运营前后的 R 值有无显著变化，来反映高铁运营是否对高速公路车辆通行压力有显著减缓作用。

运营前后的 R 值整理在下表。

表 11：高铁运营前后 R 值

	运营前 R 值	运营后 R 值	变化
R1	0.62	0.72	+0.10
R2	0.61	0.76	+0.15
R3	0.69	0.81	+0.12
R4	0.68	0.85	+0.17
R5	0.59	0.79	+0.20
R6	0.64	0.86	+0.22
R7	0.63	0.84	+0.21
R8	0.65	0.83	+0.18
R9	0.63	0.84	+0.21
R10	0.62	0.77	+0.15

使用 Excel 软件对两组数据进行显著性分析，结果如下：

表 12：显著性分析结果

SS	df	MS	F	P-value	F crit
0.019805	9	0.002201	3.178893	0.018725	2.660175

● 结论

由上表可知，P 值为 0.018725，大于 0.01，小于 0.05，说明两组数据具有显著差异，即高铁的运营对于高速公路的车辆通行压力具有显著的缓解作用。

六、 问题一的模型建立与求解

6.1 问题二的分析

高铁既快捷，又舒适，但是相对于普通列车出行价格相对昂贵，因此不同地域的人

出行方式的选择将会有所差异。我们需要选择发展水平不同的城市，分析不同城市的人口出行方式，并给出各个城市高铁配置的最佳数量。

6.2 城市发展水平分级及城市选择

新一线城市研究所对中国 338 个地级以上城市进行排名的指标为：

- 1) 商业资源聚集度；
- 2) 城市枢纽性；
- 3) 城市人活跃度；
- 4) 生活方式多样性；
- 5) 未来可塑性。

这些指标都与人们的出行方式选择相关，也具有代表性。因此我们根据城市划线分级，分别在选取了一线城市、新一线城市、二线城市、三线城市、四线城市中各选取一个为代表，对高铁配置问题进行分析，以下为选定的城市及其相关情况：

表 13：所选城市基本情况

城市名称	分级	常住人口数量	GDP	城市特点
北京	一线城市	2154.2 万人	30320 亿元	首都，枢纽城市
成都	新一线城市	1633 万人	15342.77 亿元	省会，旅游城市
济南	二线城市	870 万人	7856.56 亿元	省会，发展城市
绵阳	三线城市	485.7 万人	2303.82 亿元	科技城市
乐山	四线城市	350.68 万人	1615.09 亿元	旅游城市

6.3 人口出行方式分析

6.3.1 引入交通分配问题

在进行交通需求分析时，我们常常把城市道路系统抽象成一个计算网络。即给定无向图 $G=(V, A)$ ，其中 V 为顶点集， A 为个顶点相互连接组成的边集；网络的节点为道路的交叉点，连线为交叉点之间的路段。

交通分配问题就是在已知城市交通网络、路段的特性函数和矩阵的情况下，求网络中各路段的交通流量及交通时间。

● 不考虑路费——时间最短

对于车辆或出行者来说，如果两点之间有很多条道路的话，在不考虑花费问题时，它们显然会选择时间最短路。随着交通量的增加，最短路的走行时间也随之增加。增加到一定程度之后，最短路的走行时间会由于交通拥挤变成次短路，此时一部分出行者会转向新的最短路，同样新的最短路也会由于交通量的增加降为次短路。

● 考虑路费——引入消费水平影响系数 M

如果时间最短路所需路费超过出行者的心理预期，此时一部分出行者会转向次短

路。为了更好地衡量出行者对于路费的预期，我们引入消费水平影响系数 M 来衡量出行者的日常消费水平，其范围在 $0\sim 1$ 之间， M 的值越大，出行者越倾向于选择路费高的最短路； M 的值越小，出行者越倾向于选择路费低的次短路或其他路费更低的道路。

6.3.2 车流量——描述车辆通行

交通分配问题表现在交通流量及交通时间上，其中车流量为反映车辆通行问题的关键。车流量是由单位时间内通过某路段的车辆为标准，在一定的时间内，某条公路点上所通过的车辆数，车流量公式为：

$$\text{车流量} = \frac{\text{单位时间} \times \text{车速}}{\text{车距} + \text{车身高}} \quad (13)$$

针对动态模拟车辆在路网中选择路径的过程，下文首先建立目标规划模型，当车流量在一定容许范围内尽量大时，车辆通行能力最佳，对高铁通车后周边高速公路通行的车流量分担效果最好。

针对动态模拟车辆在路网中选择路径的过程，下文首先建立目标规划模型，当车流量在一定容许范围内尽量大时，车辆通行能力最佳，对高铁通车后周边高速公路通行的车流量分担效果最好，引入了蚁群算法求解。

蚁群算法不仅具有卓越的随机搜索寻优能力和自适应性分布式的计算特点，而且路径寻优过程与车辆选择路径过程非常相似，因此本文用蚁群算法求解交通分配问题。

6.4 目标规划模型建立

(一) 目标函数的转化与建立

目标函数为城市最佳高铁配置数量，在考虑出行者消费水平的前提下，将城市最佳高铁配置数量转化为高铁对高速公路车流量分流能力最大的问题。

目标函数即可设定为车辆通行总流量 L 最大，即高铁对高速公路车流量分流能力最大，高铁数量设置最佳， $-L$ 最小：

$$\text{Min} \quad -L = -\sum_{i=1}^n \sum_{j=1}^n (L_{ij} \cdot u_{ij}) \quad (14)$$

其中 η 为出行者出行路线改变概率， η 与出行者消费水平系数 M ($0 < M < 1$) 取值相同：

$$u_{i,j} = \begin{cases} 0, & (\eta \leq 0.5) \text{ 表示出行者不从第 } i \text{ 号坐标点到第 } j \text{ 号坐标点} \\ 1, & (\eta > 0.5) \text{ 表示出行者从第 } i \text{ 号坐标点到第 } j \text{ 号坐标点} \end{cases}$$

(二) 约束条件：

- (1) 对于各个坐标点只经过一次（即每个坐标点的进点坐标和出点坐标均是有且只有一个）：

$$\sum_{i=1}^n u_{i,t} = 1; (t = 1, 2, \dots, n) \quad (15)$$

$$\sum_{i=1}^n u_{t,i} = 1; (t = 1, 2, \dots, n) \quad (16)$$

(2) 不允许出现原地停留的现象:

$$u_{i,i} = 0; (i = 1, 2, \dots, n) \quad (17)$$

(3) 除起点和终点外, 各边不构成圈:

$$\sum_{i,j \in s} x_{ij} \leq |s| - 1, 2 \leq |s| \leq n - 1 \quad (18)$$

其中, s 为 $\{1, 2, \dots, n\}$ 的真子集。

(4) 进入节点的车流量综合与出节点的车流量总和相同:

$$x_{i,j} = u_{i,j} \left(St + \sum_{i=1}^n x_{i,j} \right); (i = 1, 2, \dots, n) \quad (19)$$

其中, ST 为到达该节点前的累计车流量。

(5) 各通行路段不允许有超过道路可通行能力的现象, 即每条道路的车流量不超过给定饱和度下的标准车流量:

$$x_{i,j} \leq N_{\text{基本通行能力}} \cdot t; (i = 1, 2, \dots, n-1; j = 1, 2, \dots, n) \quad (20)$$

(三) 目标规划模型:

在给出了目标函数和约束条件后, 即可得到一个使得最佳高铁配置数量的单目标规划模型如下:

$$\begin{aligned} \text{Min } -L = & - \sum_{i=1}^n \sum_{j=1}^n (L_{ij} \cdot u_{ij}) \\ \text{s.t. } & \begin{cases} \sum_{i=1}^n u_{i,t} = 1; (t = 1, 2, \dots, n) \\ \sum_{i=1}^n u_{t,i} = 1; (t = 1, 2, \dots, n) \\ u_{i,i} = 0; (i = 1, 2, \dots, n) \\ \sum_{i,j \in s} x_{ij} \leq |s| - 1, 2 \leq |s| \leq n - 1 \\ x_{i,j} = u_{i,j} \left(St + \sum_{i=1}^n x_{i,j} \right); (i = 1, 2, \dots, n) \\ x_{i,j} \leq N_{\text{基本通行能力}} \cdot t; (i = 1, 2, \dots, n-1; j = 1, 2, \dots, n) \end{cases} \end{aligned} \quad (21)$$

6.5 蚁群算法求解——城市高铁数量最佳配置

针对动态模拟出行者在路网中选择路径的过程, 本文引入了蚁群算法求解。蚁群算法不仅具有卓越的随机搜索寻优能力和自适应性分布式的计算特点, 而且路径寻优过程

与车辆选择路径过程非常相似，因此本文尝试用蚁群算法求解交通分配问题——即出行者出行过程中高铁分流的最大流量。

接下来即为利用蚁群优化模型求解（1，2...n 表示交叉口节点序号）。为模拟实际蚂蚁的行为，引进符号说明中解释的各种符号。

1. 人工蚁群算法演化解释：

人工蚁群的绝大部分行为特征都源于真实蚂蚁，均可以通过相互的协作在全局范围内找出问题的最优解决方案——即寻找连接起点和重点的最短路径。原则上每只蚂蚁均可建立一个解决方案，引入数量较大的蚁群，综合各个方案结果，即可得到高质量的最优方案^[9]。

在本问题中，将出行者选择个人最优路径的行为假设为人工蚁群的自发寻找最短路径行为。进行如下转化：

表 14：本问题的蚁群算法应用解释

本问题	蚁群算法解最短路径问题
出行者	人工蚁群
道路交叉口节点标号	位置标号
通行路段轨迹	蚁群选择最短路径
选择该出行方式的概率	蚁群所释放的信息素多少

2. 蚁群算法过程：

初始时刻，各条路径上信息量相等，设 $\tau_{ij}(0) = C$ （ C 为常数）。蚂蚁（车辆） k 在运动过程中，根据各条路径上信息量决定转移方向，其中蚂蚁（车辆） k 转移位置的概率 p_{ij}^k 为：

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha(t)\eta_{ij}^\alpha(t)}{\sum_{s \in allowed_k} \tau_{is}^\alpha(t)\eta_{is}^\alpha(t)}, & j \in allowed_k \\ 0, & \text{其他} \end{cases} \quad (20)$$

其中， $allowed_k = (0, 1, \dots, n-1) - tabu_k$ 为蚂蚁（车辆） k 下一步允许选择的位置（道路交叉口节点）标号。

与实际蚁群不同，人工蚁群（车辆）系统具有记忆功能， $tabu_k (k = 1, 2, \dots, m)$ 用以记录蚂蚁（车辆） k 以前走过的城市，集合 $tabu_k$ 随着进化过程做动态调整。

随着时间的推移，以前留下的信息逐渐消逝，用参数 $1 - \rho$ 表示信息挥发程度，经过 n 个时刻，蚂蚁（车辆）完成一次循环，各路径上信息量要根据下式调整为^[10]：

$$\tau_{ij}(t+n) = \rho * \tau_{ij}(t) + \Delta \tau_{ij} \quad \rho \in (0, 1) \quad (21)$$

其中 $\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k$ ， τ_{ij}^k 表示第 k 只蚂蚁（车辆）在本次循环中留在路径上的信息量， $\Delta\tau_{ij}$ 表示本次循环中路径 (i, j) 上的信息量增量。当一条路径上计算得到的 p_{ij}^k 最大时，即可令 $u_{ij} = 1$ 。

在 Dorigo 给出的蚁周模型中：

$$\Delta\tau_{ij} = \begin{cases} \frac{Q}{L_k}, & \text{若第 } k \text{ 只蚂蚁（车辆）在本次循环中经过路径 } (i, j) \\ 0, & \text{否则} \end{cases}$$

利用的是整体信息，在求解最短路问题时蚁周模型性能较好，因而通常用它作为基本模型。基本蚁群算法中参数可以用试验方法确定其最优组合。停止条件可以用固定进化代数或者当进化趋势不明显时便停止计算。该算法复杂度为 $O(nc \cdot n^3)$ ，其中 nc 表示循环次数。

3. ACS 局部更新规则及全局更新：

局部更新规则的应用是的相应的信息素轨迹量逐渐减少，ACS 局部更新原则使得曾经被访问过的路径在被蚂蚁访问的时候越来越缺乏吸引力，间接地使得蚂蚁更倾向于搜索还未被访问过的路径。从而，蚂蚁不会收敛到同一路径上，及会有更高的概率可以搜索到更优的解。

通过这种方法，蚂蚁可以更好地利用信息素，优于没有局部更新的蚁群系统。也就是说，基于局部更新规则的蚁群算法，再进行全局的信息素更新，即可得到最优解，即为最短路径。

6.6 模型求解

利用 C 语言进行编程求解最短路径的程序见附录二，通过资料查询和反复的迭代实验，将程序中的重要参数设置如下：

表 15: 参数设置

参数名称	NcMax	α	β	γ	α_1	ρ
代入值	1000	2	5	0.1	0.1	0.1

本问题使用 Visual C++6.0 软件进行程序设计，实现蚁群算法的求解。为使程序方便阅读，给予其模块化设计。算法流程图如下：

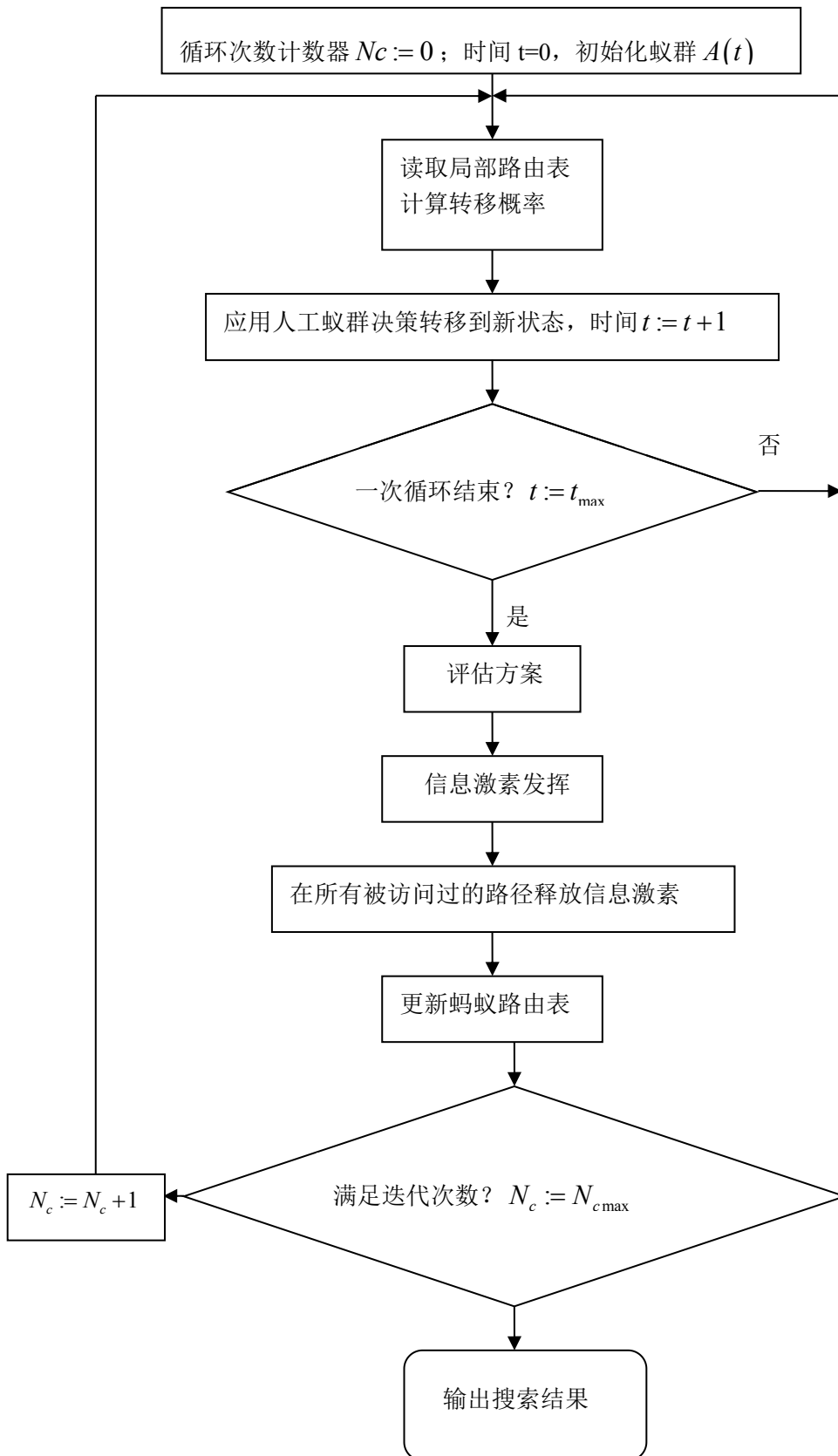


图 1: 蚁群算法的优化流程图

求解结果如下:

表 16：所选城市最佳高铁配置数量

城市名称	高铁站数量	最佳高铁配置数量
北京	5	62
成都	2	48
济南	3	39
绵阳	1	22
乐山	1	13

七、模型评价

9.1 模型优点

1、利用层次分析方法通将定性方法与定量方法有机地结合起来，把复杂目标、多个准则而且难以经过量化处理的决策问题转化为简单目标层次问题，能够直观简单的得到相对理想方案的权重向量。

2、结合指标评价的边界不十分明显的特性，避免将指标单纯归于某个类别，于是我们对每个因素进行评分划区间，然后对所有对象结合因素进行综合模糊评价，防止遗漏任何统计信息和信息的中途损失。

3、人们也采用神经网络、遗传算法和模糊等方法推算各路段交通流量。但是，假定用户在路径选择方面拥有完备信息等，使得这些算法并不能比较真实动态的模拟车辆在路网中选择路径的过程，分配结果与实际有一定差距。但蚁群算法真实性较高。

4、通过对固定需求交通平衡模型的研究发现，平衡分配问题的计算具有一定的随机性和并行性的特点，而蚁群算法不仅具有卓越的随机搜索寻优能力还具有自适应性分布式的计算特点，即它具有群体合作、正反馈和并行性等特点，而且蚁群算法的路径寻优过程与车辆选择路径过程非常相似，能使分配结果更符合实际，并逐步趋于平衡。

9.2 模型缺点

1、层次分析法的判断矩阵与模糊矩阵的建立主观因素所占比例较大，所利用的数据定性因素成分较大；

2、模糊模型构建简单，对指标权重矢量的确定主观性较强；

3、需要明确各个节点之间的具体参数，例如距离、时间等，模型前部分处理相对复杂。

八、参考文献

- [1] 蔡建荣.交通网络节点终重要度评估方法研究及其应用.长沙理工大学.2014.05.
- [2] 贾晓敏.城市道路通行能力影响因素研究.长安大学.2009.04.
- [3] 姬海.城市道路网容量节点路段化分析方法研究.哈尔滨工业大学.2010.07.

- [4] 秦葛.城市封闭式居住小区规划模式与城市交通发展的协调性研究.长安大学.2010.04.
- [5] 郭跃华.陆志峰.王建宏.城市交叉路口汽车通行模型.南通大学理学院.Vol.10 No.20 July 2010.
- [6] 李向朋.城市交通拥堵对策—封闭型小区交通开放研究.长沙理工大学.2014.04.
- [7] 张海明.城市居住片区交通微循环系统研究.西安建筑科技大学.2011.05.
- [8] 陈春安.荣建.周晨静.常鑫.分_合流区通行能力分析方法改进研究.北京工业大学.Vol.15 No.6 Dec.2015.
- [9] 方丽君.基于蚁群算法的交通分配模型研究.河海大学.2006.05.
- [10] 徐纪锋.基于蚁群算法的交通网络优化方法研究.山东理工大学.2008.12.
- [11] 满意度数学建模 <http://wenku.baidu.com/view/bd42157b312b3169a451a4bd.html>
- [12] 廖红强,邱勇,杨侠,杨星刚,葛任伟.对应用层次分析法确定权重系数的探讨.2012年第六期;

九、 附录

附录 1: 层次分析法 Matlab 程序

```

clc;
clear;
A = [1 7/9 7/4
     9/7 1 9/4
     4/7 4/9 1];
[m, n]=size(A);           %获取指标个数
RI=[0 0 0.58 0.90 1.12 1.24 1.32 1.41 1.45 1.49 1.51];
R=rank(A);                %求判断矩阵的秩
[V, D]=eig(A);           %求判断矩阵的特征值和特征向量, V 特征值, D 特征
向量;
tz=max(D);
B=max(tz);                %最大特征值
[row, col]=find(D==B);   %最大特征值所在位置
C=V(:, col);              %对应特征向量
CI=(B-n)/(n-1);          %计算一致性检验指标 CI
CR=CI/RI(1, n);
if CR<0.10
    disp('CI=');disp(CI);
    disp('CR=');disp(CR);
    disp('对比矩阵 A 通过一致性检验, 各向量权重向量 Q 为: ');
    Q=zeros(n, 1);
    for i=1:n
        Q(i, 1)=C(i, 1)/sum(C(:, 1));%特征向量标准化
    end
end

```

```

        end
        Q                                %输出权重向量
    else
        disp('对比矩阵 A 未通过一致性检验，需对对比矩阵 A 重新构造');
    end
end

```

附录 2：C 语言程序

```

#include<iostream>
#include<math.h>
#include<time.h>
using namespace std;
//城市节点数目
#define N 12
//城市坐标
double C[N][2]={
    {13,13},{68.5,13},{123.5,13},{13,43.5},{68.5,43.5},{123.5,43.5},
    {13,58.5},{68.5,58.5},{123.5,58.5},{13,78.5},{68.5,78.5},{123.5,78.5}};
//蚂蚁数量
#define M 12
//最大循环次数 NcMax
int NcMax =1000;
double alpha = 2, beta = 5, rou = 0.1, alpha1= 0.1, qzero = 0.1;
double Lnn;
//矩阵表示两两城市之间的距离
double allDistance[N][N];
//计算两个城市之间的距离
double calculateDistance(int i, int j)
{
    return sqrt(pow((C[i][0]-C[j][0]),2.0) + pow((C[i][1]-C[j][1]),2.0));
}
void calculateAllDistance()
{
    for(int i = 0; i < N; i++)
    {
        for(int j = 0; j < N; j++)
        {
            if (i != j)
            {
                allDistance[i][j] = calculateDistance(i, j);
                allDistance[j][i] = allDistance[i][j];
            }
        }
    }
}
//获得经过 n 个城市的路径长度

```



```

double calculateSumOfDistance(int* tour)
{
    double sum = 0;
    for(int i = 0; i < N ; i++)
    {
        int row = *(tour + 2 * i);
        int col = *(tour + 2 * i + 1);
        sum += allDistance[row][col];
    }
    return sum;
}
class ACSAnt;
class AntColonySystem
{
private:
    double info[N][N], visible[N][N]; //节点之间的信息素强度,节点之间的能见度
public:
    AntColonySystem()
    {
    }
//计算当前节点到下一节点转移的概率
    double Transition(int i, int j);
//局部更新规则
    void UpdateLocalPathRule(int i, int j);
//初始化
    void InitParameter(double value);
//全局信息素更新
    void UpdateGlobalPathRule(int* bestTour, int globalBestLength);
};
//计算当前节点到下一节点转移的概率
double AntColonySystem::Transition(int i, int j)
{
    if (i != j)
    {
        return (pow(info[i][j], alpha) * pow(visible[i][j], beta));
    }
    else
    {
        return 0.0;
    }
}
//局部更新规则
void AntColonySystem::UpdateLocalPathRule(int i, int j)
{
    info[i][j] = (1.0 - alpha1) * info[i][j] + alpha1 * (1.0 / (N * Lnn));
}

```

```

        info[j][i] = info[i][j];
    }
    //初始化
    void AntColonySystem::InitParameter(double value)
    {
        //初始化路径上的信息素强度 tao0
        for(int i = 0; i < N; i++)
        {
            for(int j = 0; j < N; j++)
            {
                info[i][j] = value;
                info[j][i] = value;
                if (i != j)
                {
                    visible[i][j] = 1.0 / allDistance[i][j];
                    visible[j][i] = visible[i][j];
                }
            }
        }
    }
    //全局信息素更新
    void AntColonySystem::UpdateGlobalPathRule(int* bestTour, int globalBestLength)
    {
        for(int i = 0; i < N; i++)
        {
            int row = *(bestTour + 2 * i);
            int col = *(bestTour + 2 * i + 1);
            info[row][col] = (1.0 - rou) * info[row][col] + rou * (1.0 / globalBestLength);
            info[col][row] = info[row][col];
        }
    }
class ACSAnt
{
private:
    AntColonySystem* antColony;
protected:
    int startCity, cururentCity;//初始城市编号，当前城市编号
    int allowed[N];//禁忌表
    int Tour[N][2];//当前路径
    int currentTourIndex;//当前路径索引，从 0 开始，存储蚂蚁经过城市的编号
public:
    ACSAnt(AntColonySystem* acs, int start)
    {
        antColony = acs;
        startCity = start;
    }

```

```

}
//开始搜索
int* Search();
//选择下一节点
int Choose();
//移动到下一节点
void MoveToNextCity(int nextCity);
};
//开始搜索
int* ACSAnt::Search()
{
    cururentCity = startCity;
    int toCity;
    currentTourIndex = 0;
    for(int i = 0; i < N; i++)
    {
        allowed[i] = 1;
    }
    allowed[cururentCity] = 0;
    int endCity;
    int count = 0;
    do
    {
        count++;
        endCity=cururentCity;
        toCity=Choose();
        if(toCity>=0)
        {
            MoveToNextCity(toCity);
            antColony->UpdateLocalPathRule(endCity,toCity);
            cururentCity=toCity;
        }
    }while(toCity>=0);
    MoveToNextCity(startCity);
    antColony->UpdateLocalPathRule(endCity,startCity);
    return*Tour;
}
//选择下一节点
int ACSAnt::Choose()
{
    int nextCity=-1;
    double q=rand()/(double)RAND_MAX;
    //如果 q<=q0,按先验知识, 否则则按概率转移,
    if (q<=qzero)
    {

```

```

double probability = -1.0;//转移到下一节点的概率
for(int i = 0; i < N; i++)
{
//去掉禁忌表中已走过的节点,从剩下节点中选择最大概率的可行节点
    if (1 == allowed[i])
    {
        double prob = antColony->Transition(cururentCity, i);
        if (prob > probability)
        {
            nextCity = i;
            probability = prob;
        }
    }
}
else
{
//按概率转移
double p = rand()/(double)RAND_MAX;//生成一个随机数,用来判断落在哪个区间段
double sum = 0.0;
double probability = 0.0;//概率的区间点, p 落在哪个区间段, 则该点是转移的方向
//计算概率公式的分母的值
for(int i = 0; i < N; i++)
{
    if (1 == allowed[i])
    {
sum += antColony->Transition(cururentCity, i);
    }
}
for(int j = 0; j < N; j++)
{
    if (1 == allowed[j] && sum > 0)
    {
        probability += antColony->Transition(cururentCity, j)/sum;
        if (probability >= p || (p > 0.9999 && probability > 0.9999))
        {
            nextCity = j;
            break;
        }
    }
}
}
return nextCity;
}
//移动到下一节点

```

```

void ACSAnt::MoveToNextCity(int nextCity)
{
allowed[nextCity]=0;
Tour[currentTourIndex][0] = currentCity;
Tour[currentTourIndex][1] = nextCity;
currentTourIndex++;
currentCity = nextCity;
}
//-----
//选择下一个节点，配合下面的函数来计算长度
int ChooseNextNode(int currentNode, int visitedNode[])
{
    int nextNode = -1;
    double shortDistance = 0.0;
    for(int i = 0; i < N; i++)
    {
        //去掉已走过的节点,从剩下节点中选择距离最近的节点
        if (1 == visitedNode[i])
        {
            if (shortDistance == 0.0)
            {
                shortDistance = allDistance[currentNode][i];
                nextNode = i;
            }
            if(shortDistance < allDistance[currentNode][i])
            {
                nextNode = i;
            }
        }
    }
    return nextNode;
}
//给一个节点由最近邻距离方法计算长度
double CalAdjacentDistance(int node)
{
    double sum = 0.0;
    int visitedNode[N];
    for(int j = 0; j < N; j++)
    {
        visitedNode[j] = 1;
    }
    visitedNode[node] = 0;
    int currentNode = node;
    int nextNode;
    do
    {

```

```

    nextNode = ChooseNextNode(currentNode, visitedNode);
    if (nextNode >= 0)
    {
        sum += allDistance[currentNode][nextNode];
        currentNode = nextNode;
        visitedNode[currentNode] = 0;
    }
} while(nextNode >= 0);
sum += allDistance[currentNode][node];
return sum;
}
//-----结束-----
//-----主函数-----
int main()
{
    time_t timer, timer1;
    time(&timer);
    unsigned long seed = timer;
    seed %= 56000;
    srand((unsigned int)seed);
    //由矩阵表示两两城市之间的距离
    calculateAllDistance();
    //蚁群系统对象
    AntColonySystem* acs = new AntColonySystem();
    ACSAnt* ants[M];
    //蚂蚁均匀分布在城市上
    for(int k = 0; k < M; k++)
    {
        ants[k] = new ACSAnt(acs, (int)(k%N));
    }
    calculateAllDistance();
    int node = rand() % N;
    Lnn = CalAdjacentDistance(node);
    double initInfo = 1 / (N * Lnn);
    acs->InitParameter(initInfo);
    int globalTour[N][2];
    double globalBestLength = 0.0;
    for(int i = 0; i < NCMAX; i++)
    {
        int localTour[N][2];
        double localBestLength = 0.0;
        double tourLength;
        for(int j = 0; j < M; j++)
        {
            int* tourPath = ants[j]->Search();

```

```

tourLength = calculateSumOfDistance(tourPath);
//局部比较，并记录路径和长度
if(tourLength < localBestLength || abs(localBestLength - 0.0) < 0.000001)
{
    for(int m = 0; m < N; m++)
    {
        int row = *(tourPath + 2 * m);
        int col = *(tourPath + 2 * m + 1);
        localTour[m][0] = row;
        localTour[m][1] = col;
    }
    localBestLength = tourLength;
}
}
if(localBestLength < globalBestLength || abs(globalBestLength - 0.0) < 0.000001)
{
    for(int m = 0; m < N; m++)
    {
        globalTour[m][0] = localTour[m][0];
        globalTour[m][1] = localTour[m][1];
    }
    globalBestLength = localBestLength;
}
acs->UpdateGlobalPathRule(*globalTour, globalBestLength);
cout<<"第 " << i + 1 << " 迭代最优路径:" << localBestLength << ". " << endl;
for(int m = 0; m < N; m++)
{
    cout<<localTour[m][0]<<". ";
}
cout<<endl;
}
cout<<"全局最优路径长度:" << globalBestLength << endl;
cout<<"全局最优路径:";
for(int m = 0; m < N; m++)
{
    cout<<globalTour[m][0]<<". ";
}
cout<<endl;
time(&timer1);
int t = timer1 - timer;
return 0;
}

```