

城市相邻两交叉口信号配时优化问题

摘要

本文使用了模糊折中的多目标规划模型、Webster 算法、遗传算法、蚁群算法等方法解决了城市相邻两交叉口信号配时优化问题。

论文从两个方面对交通信号进行配时优化研究，分别是单交叉口相位时间配时和相邻交叉口相位差优化分析。

对于单交叉口相位时间配时方案的确定，文章从典型四相位系统出发，首先根据黄灯问题，计算出最优的黄灯配时为 $3s$ 。其次，以影响单点信号配时的三个主要因素（车辆延误时间、平均停车次数、交叉口的通行能力）为优化对象，建立各因素的配时优化模型，采用模糊折中规划的多目标规划方法，将这三个目标函数转化为一个单目标函数。其中，使用模糊偏好的方法，确定三者的隶属度函数在单目标函数中的权重。最后分别使用 Webster 算法、遗传算法、蚁群算法进行优化求解，得到各个算法计算出来的有效绿灯时间，并将其代入原目标函数进行对比。并且从时间复杂度、空间复杂度、收敛性三个方面对遗传算法和蚁群算法进行了对比分析。基于以上因素得到最终的单点相位时间配时方案，其中 A 交叉路口的各相位配时为 54,16,24,25，B 交叉路口的各相位配时 44,22,34,14。

对于相邻两交叉口相位差方案的分析，将上述建立的三个主要的影响因素优化模型进行修改，得到了影响相邻交叉口相位差的车辆延误模型、平均停车次数模型，交叉口的通行能力模型。然后，采用统一目标函数法将多目标优化问题转化为单目标最小化问题。在假设 A、B 两个路口之间的距离为 500m，车流的速度为 $36km/h$ 的前提下，通过遗传算法得到两个相邻交叉口的相位差为 $9s$ 。在得到优化结果后，本文又进行了仿真检验，得到车均延误时间没有规律性的变化而且波动性很大的结论。

本文综合模糊折中的多目标规划模型，将多目标函数单一化处理，采用 Webster 算法、遗传算法、蚁群算法求解出相邻两交叉口信号配时优化问题。结合 Matlab、Vissim 等相关软件，对城市相邻两交叉口信号配时优化问题进行了全面的分析，并给出了题目中 A、B 两个交叉口的配时方案。在文章的最后对模型的适用范围做出了推广，在实际应用中有一定的参考价值。

关键词：多目标规划 Webster 算法 遗传算法 蚁群算法 典型四相位



目录

一、 问题重述	3
1.1 问题背景	3
1.2 问题复述	4
二、 问题分析	4
2.1 单交叉口相位时间配时方案的分析	4
2.2 相邻两交叉口相位差方案的分析	4
三、 模型假设	4
四、 符号说明	4
五、 模型建立与求解	6
5.1 单点信号配时问题的模型建立与求解	6
5.1.1 黄灯问题	6
5.1.2 信号配时模型的构建	8
5.1.3 模糊折中规划的模型建立	10
5.1.4 Webster 算法的模型求解	13
5.1.5 遗传算法的模型求解	14
5.1.6 蚁群算法的模型求解	17
5.1.7 不同算法的对比分析及最终方案的确定	19
5.2 相邻交叉口相位差优化模型的建立与求解	21
5.2.1 相位差和周期	21
5.2.2 车辆延误模型	21
5.2.3 平均停车次数	21
5.2.4 交叉口通行能力	21
5.2.5 优化模型及其求解	22
5.3 仿真检验	23
六、 模型评价	25
6.1 模型优点	25
6.2 模型缺点	26
6.3 模型推广	26
七、 参考文献	26
八、 附录	27
8.1 遗传算法的程序	27
8.2 蚁群算法的程序	27



一、问题重述

1.1 问题背景

随着社会的飞速发展，城市的交通直接反映了一个城市的文明程度。然而，城市化进程不断加快，我国各个城市的机动车的数量在急剧上升，导致机动车的增长速率远远地高于新建道路的增长率，致使城市的交通需求量与供给需求量严重不匹配。尤其是在高峰时段，经常出现堵车，车辆行驶缓慢的现象。车辆拥堵严重，由此导致了一系列诸如出行成本提高、空气污染加剧、交通安全隐患等问题。

为了提高城市道路交通管理，改善人们的出行条件，保证公路交通的畅通与安全，当今世界各国交通系统规模不断扩大，已逐渐从单叉口信号的点控和单条干道的线控，发展到整个交通网络的面控。世界各地广泛使用，智能交通系统，而此系统的核心问题是交通信号的智能控制。目前我国的大中型城市交通管理普遍采用单点定时交通信号灯控制。而这种传统信号灯控制方法会造成某些方向的绿时浪费和车辆延误现象。

1.2 问题复述

在题目已给的背景下，以武汉市某相邻的两个交叉口 A, B 为例，在已知的数据前提下，我们建立数学模型讨论下列问题。

- 1.设计通用模型与算法(从时间复杂度、空间复杂度、收敛性进行对比分析)，对交通信号进行配时优化研究，
- 2.求解出改善后的交通信号配时方案并进行仿真检验。

二、问题分析

该问题要求设计通用模型与算法，对交通信号进行配时优化，针对相邻两交叉口 A,B,可从两方面考虑，其一为单交叉口相位时间配时方案研究，其二为相邻两交叉口相位差的方案确定。

2.1 单交叉口相位时间配时方案的分析

单交叉口的信号优化一般以延误时间最小作为唯一优化目标，显然这样考虑的因素太少，得到的目标函数的结果并非最优。其中，衡量交叉口运行效率的主要因素为总延误、停车次数、通行能力，故可建立多个目标函数进行优化，即可将其归纳为多目标规划问题。

针对多目标规划问题，可利用遗传算法和蚁群算法进行求解，首先对两种算法进行时间复杂度、空间复杂度、收敛性的对比分析，其次对于两种求解结果可将其带入原目标函数进行对比。基于以上所有因素进行综合考虑，我们可以确定出最终的单交叉口相位时间配时方案。

2.2 相邻两交叉口相位差方案的分析

在城市道路网中，如果各交叉口相距很近，为了保证车辆的畅通行驶，两路口间必须有一定的相位差。其中相位差一般指两邻近交叉口相同相位的绿灯开始时间的间隔。针对该问题依然可以将总延误、停车次数、通行能力作为优化目标，基于单交叉口问题建立该问题的多目标规划，最终可以利用遗传算法进行求解。对于求解结果可算出总延误、停车次数、通行能力与原方案进行对比分析。基于以上分析，可以确定出 A、B 两交叉口各自的相位时间配时方案，以及两者之间的相位差方案。最终，针对此问题可以利用 VISSIM 软件进行仿真，得到仿真结果。

三、模型假设

- 1.假设不考虑非机动车辆和行人对交通的影响。由于在实际生活中非机动车辆和行人对交通是存在一定的影响。
- 2.假设交通流量在研究的过程中是基本不变的。由于随着时间点的不同，交通流量存在着较大的差异。
- 3.假设各个车道上没有专用的特殊车道，否则会极大的影响车流的速度。
- 4.假设各个方向的车道在交叉口右转的时候，不需要看红绿灯。

四、符号说明

符号	符号说明
L	车辆在行驶过程中向前行驶的距离
L_1	司机刹车反应时间驶过的距离
L_2	刹车制动后车辆驶过的距离
t_1	司机反应时间
v	路口限速
t_2	刹车制动时间
D	十字路口的距离
T	黄灯应亮时间
q_i	车道组 i 的到达率
C	表示信号周期
d_i	车道组 i 的每辆车平均停车延误
u_i	表示车道组 i 的绿信比

y_i	表示车道组 i 的流量比
g_i	表示车道组 i 所属相位的有效绿灯时间
l_i	表示车道组 i 损失时间
h_i	表示相位 i 的停车率

五、模型建立与求解

5.1 单点信号配时问题的模型建立与求解

此问题要求对四相位的单点信号配时问题进行优化，得到对于单点信号情况下的最优配时问题。我们认为有以下五个步骤：

步骤一：首先在 Webster 算法[1]的基础上，分析出影响单点信号配时的三个主要因素，然后建立各个因素的配时优化模型，其目的是得到相应的优化目标函数。

步骤二：将多目标函数化为单目标函数，基于模糊折中规划的多目标规划的基本思想，使得到的各个函数对单目标函数的影响由权重系数决定。

步骤三：使用 Webster 算法进行优化求解。

步骤四：再使用蚁群算法和遗传算法对所得到的单目标函数进行优化求解。

步骤五：对所使用的通用算法进行时间复杂度、空间复杂度、收敛性进行对比分析，最后求解出改善后的单点信号配饰方案。

5.1.1 黄灯问题

由于，根据相关文献报道，发生在黄灯时间内的交通事故约占该交叉口处所有交通事故的一半以上。因此，黄灯时间的长短对于改善交通拥堵和降低交通事故发生概率是十分重要的。为了使模型在实际问题中有更大的使用价值，在本文中，针对黄灯问题进行了具体的讨论。

研究这个问题，应考虑两方面，一方面，黄灯亮起，司机若能停车应立即停车，而停车需要时间，该时间由于司机的反应时间和汽车制动时间有关，在这段时间内汽车将继续行驶一段距离；另一方面，对于黄灯亮起已过线的车辆应当保证他们仍能穿过马路。综上分析，可将黄灯应亮的时间分为两部分，一部分为停车时间，另一部分为过路口时间。

首先考虑，司机在看到黄灯亮起时，刹车减速应行驶多长距离车才能停下来，这需要考虑司机的反应时间 t_1 ，其长短与很多因素有关，例如司机的驾龄，年龄，性别以及时速等。据统计，85%的司机反映时间为 0.3–1.0s，根据多方面的考虑我们取反应时间为 0.5s。假设车辆在行驶过程中向前行驶的距离为 L ，也就是说，在离街口距离为 L 处有一条停车线， L 应分为两段，即：

$$L = L_1 + L_2 \quad (1)$$

其中 L_1 表示司机刹车反应时间驶过的距离， L_2 表示刹车制动后车辆驶过的距离。假设该路段的限速为每辆汽车行驶的速度，并假设汽车都以速度 v 匀速运动，即：

$$L_1 = t_1 \times v \quad (2)$$

假设汽车只受摩擦力的作用，且各路段的摩擦系数相同，则汽车刹车制动力为：

$$F = \mu mg \quad (3)$$

根据牛顿第二定律，距离 L_2 为：

$$\begin{cases} ma = \mu mg \\ v = at_2 \\ L_2 = \frac{1}{2} at_2^2 \end{cases} \quad (4)$$

黄灯应亮时间为：

$$T = \frac{L + D}{v} \quad (5)$$

综上可建立数学模型为：

$$\begin{cases} L = L_1 + L_2 \\ L_1 = t_1 \times v \\ F = \mu mg \\ ma = \mu mg \\ v = at_2 \\ L_2 = \frac{1}{2} at_2^2 \\ T = \frac{L + D}{v} \end{cases} \quad (6)$$

$$\text{解得： } T = \frac{D}{v} + t_1 + \frac{v}{2ug} \quad (7)$$

为了简化计算，在这里我们取道路宽度为 20 米，道路限速为 40 公里，反应时间为 0.5 秒，摩擦系数根据表 1 取平均值，重力加速度为 10，经计算得黄灯大约应亮 3 秒。

表 1 车辆轮胎与各种路面间的摩擦系数

路面类型	滚动阻力系数
滚动阻力系数良好的沥青或混凝土路面	0.010-0.018
一般的；沥青或混凝土路面	0.018-0.020
碎石路面	0.020-0.025
良好的卵石路面	0.025-0.030
坑洼的卵石路面	0.035-0.050
压紧土路，干燥的	0.025-0.035
雨后的泥泞土路	0.050-0.150
干沙	0.100-0.300
湿沙	0.060-0.150
结冰路面	0.015-0.030
压紧的雪道	0.030-0.050

5.1.2 信号配时模型的构建

由于对于交叉口的配时模型，本文将其分为交叉口的总延误[2]，总停车次数[3]，总通行能力[4]这三个目标作为优化对象，采用了模糊折中的规划方法[5]对这三个目标进行无量纲化处理[6]，

交叉口的总延误是指由于交叉口处信号控制引起交通流间断而损失的车辆行驶的总时间。其中平均延误时间如下：

$$d_i = \frac{C(1.0 - u_i)^2}{2(1.0 - y_i)} \quad (8)$$

其中，Webster 最佳周期计算公式为：

$$C = \frac{1.5L + 5.0}{1.0 - Y} \quad (9)$$

其中 L 为交叉口损失时间(s), Y 为交叉口流率比。

假设交叉信号有 n 个相位，则一个周期内交叉口的总延误的最小值（用 A 表示），根据交叉口的总延误的定义得到如下的模型：

$$\begin{aligned}
\min A &= \sum_{i=1}^n q_i C d_i = \sum_{i=1}^n q_i C \frac{C(1-u_i)^2}{2(1-y_i)} = \sum_{i=1}^n \frac{q_i C^2 (1-u_i)^2}{2(1-y_i)} \\
&= \sum_{i=1}^n \frac{q_i C^2 \left(1 - \frac{g_i}{C}\right)^2}{2(1-y_i)} = \sum_{i=1}^n \frac{q_i \left[C \left(1 - \frac{g_i}{C}\right)\right]^2}{2(1-y_i)} \\
&= \sum_{i=1}^n \frac{q_i \left\{ \sum_{i=1}^n (g_i + l_i) \left[1 - \frac{g_i}{\sum_{i=1}^n (g_i + l_i)} \right] \right\}^2}{2(1-y_i)} \tag{10}
\end{aligned}$$

其中, q_i 表示车道组 i 的到达率, C 表示信号周期, d_i 表示车道组 i 的每辆车平均停车延误, u_i 表示车道组 i 的绿信比, y_i 表示车道组 i 的流量比, g_i 表示车道组 i 所属相位的有效绿灯时间, l_i 表示车道组 i 损失时间。车辆的平均停车次数计算公式为:

$$h = 0.9 \left(\frac{1.0 - u}{1.0 - y} \right) \tag{11}$$

一个周期内交叉口的总停车次数的最小值 (用 B 表示) 可以表示为

$$\begin{aligned}
\min B &= \sum_{i=1}^n h_i q_i C = \sum_{i=1}^n 0.9 \left(\frac{1-u_i}{1-y_i} \right) q_i C = \sum_{i=1}^n 0.9 \left(\frac{1 - \frac{g_i}{C}}{1-y_i} \right) q_i C \\
&= \sum_{i=1}^n 0.9 q_i \left(\frac{1 - \frac{g_i}{\sum_{i=1}^n (g_i + l_i)}}{1-y_i} \right) \left(\sum_{i=1}^n (g_i + l_i) \right) \tag{12}
\end{aligned}$$

其中, h_i 表示相位 i 的停车率。

一个周期内交叉口的通行能力为:

$$Q_i = s_i (x_i / c), \tag{13}$$

则总通行能力的相反数最小值 (用 C 表示) 可以表示为

$$\min C = \sum_{i=1}^n (-Q_i) = \sum_{i=1}^n (-S_i u_i) = \sum_{i=1}^n \left(-S_i \left(\frac{g_i}{C} \right) \right) = \sum_{i=1}^n \frac{-S_i g_i}{\left[\sum_{i=1}^n (g_i + l_i) \right]} \quad (14)$$

三个目标函数的约束条件为：

$$s.t. \begin{cases} g_i \geq g_{i \min}, 1 \leq i \leq n \\ \sum_{i=1}^n (g_i + l_i) \leq C_{\max} \\ u_1 \leq -\frac{y_i}{g_i/C} \leq u_2 \end{cases} \quad (15)$$

最后将各个目标函数化为一个形如式子（16）的单目标函数进行求解

$$f = f(A, B, C) \quad (16)$$

5.1.3 模糊折中规划的模型建立

在多目标规划信号配时中，本文使用模糊折中规划的方法，其具体步骤如下：

步骤一：将上述（10）（12）（14）三个标准形式的式子在各自的约束条件

下，由最小值组成的理想值向量 Y_{\min} ：

$$Y_{\min} = (y_1, y_2, y_3) = (A_{\min}, B_{\min}, C_{\min}) \quad (17)$$

并且求出各个单目标函数在相应的约束条件下，由最大值组成的反理想值向量 Y_{\max} ：

$$Y_{\max} = (y_1, y_2, y_3) = (A_{\max}, B_{\max}, C_{\max}) \quad (18)$$

步骤二：按照如下的方式定义各个单目标函数的隶属度函数[7]：

$$U = \begin{cases} 1 & Y \leq Y_{\min} \\ \frac{Y_{\max} - Y_i}{Y_{\max} - Y_{\min}} & Y_{\min} < Y_i < Y_{\max} \\ 0 & Y_{\max} \leq Y_i \end{cases} \quad (19)$$

$$U = (u_i)_{1 \times 3} \quad (20)$$

则 A 的隶属度函数为：

$$u_1 = \begin{cases} 1 & A \leq A_{\min} \\ \frac{A_{\max} - A}{A_{\max} - A_{\min}} & A_{\min} < A < A_{\max} \\ 0 & A_{\max} \leq A \end{cases} \quad (21)$$

B 的隶属度函数为：

$$u_2 = \begin{cases} 1 & B \leq B_{\min} \\ \frac{B_{\max} - B}{B_{\max} - B_{\min}} & B_{\min} < B < B_{\max} \\ 0 & B_{\max} \leq B \end{cases} \quad (22)$$

C 的隶属度函数为:

$$u_3 = \begin{cases} 1 & C \leq C_{\min} \\ \frac{C_{\max} - C}{C_{\max} - C_{\min}} & C_{\min} < C < C_{\max} \\ 0 & C_{\max} \leq C \end{cases} \quad (23)$$

步骤三: 将上述得到的多目标函数转换为单目标函数:

$$L = \max \left[\sum_{i=1}^3 (\lambda_i u_i)^p \right]^{\frac{1}{p}} \quad (24)$$

步骤四: 采用模糊偏好的方法[8]计算权重系数 λ_i :

对于某一多目标优化问题, 关于目标函数的性状理解往往不是很深, 通常难以确定各个目标函数的权重。因此, 目标函数线性加权法的应用受到了很大的限制。为此常常需要定义一组关系来确定各个目标函数的权重, 这组关系成为偏好关系, 下面是信号配时中各个优化目标函数的权重计算的模型建立过程。

(1) 函数分组

将所得到的三个目标函数按照重要程度分为重要和不重要两组, 然而, 对于不重要的一组, 为了计算的方便, 令其权重为零, 而重要的一组用函数 F 表示

$$F = \{f_i\}, i = 1, 2, 3, \dots \quad (25)$$

由于, 在本文中我们主要讨论的是交叉口的总延误、总停车次数、总通行能力这三个因素, 因此, 都归为重要因素一类, 所以得到:

$$F = \{u_1, u_2, u_3\} \quad (26)$$

(2) 构造等价类

用关系 \approx 构造等价类, 从每一个等价类中选择一个元素构成集合 T

$$T = \{t_i\}, i = 1, 2, 3, \dots \quad (27)$$

(3) 使用评价函数 v 给出关系的值

$$\begin{cases} v(t_i) = \alpha, v(t_j) = \beta (t_i < t_j) \\ v(t_i) = v(t_j) = \gamma \quad (t_i \approx t_j) \\ v(t_i) = \delta, v(t_j) = \varepsilon (t_i << t_j) \end{cases} \quad (28)$$

其中 $\alpha, \beta, \gamma, \delta, \varepsilon$ 为 (0,1) 之间的实数, 且 $\alpha + \beta = \delta + \varepsilon = 1$, 一般取

$\delta < \alpha < \gamma < \varepsilon < \beta$ 。

(4) 构造矩阵 R_a

根据上述的关系值，并且结合（24）式，构造出了一个 i 阶的方阵 R_a

$$\begin{cases} R_a(i, j) = 0, R_a(j, i) = 2 & (t_i \ll t_j) \\ R_a(i, j) = 0, R_a(j, i) = 1 & (t_i < t_j) \\ R_a(i, j) = 1, R_a(j, i) = 1 & (t_i \approx t_j) \end{cases} \quad (29)$$

(5) 构造偏好关系矩阵 R

结合(25)式，构造出一个 i 阶的方阵 R

$$\begin{cases} R(i, j) = \gamma, R(j, i) = \gamma & (R_a(i, j) = 1 \text{ 且 } R_a(j, i) = 1) \\ R(i, j) = \alpha, R(j, i) = \beta & (R_a(i, j) = 0 \text{ 且 } R_a(j, i) = 1) \\ R(i, j) = \beta, R(j, i) = \alpha & (R_a(i, j) = 1 \text{ 且 } R_a(j, i) = 0) \\ R(i, j) = \delta, R(j, i) = \varepsilon & (R_a(i, j) = 0 \text{ 且 } R_a(j, i) = 2) \\ R(i, j) = \varepsilon, R(j, i) = \delta & (R_a(i, j) = 2 \text{ 且 } R_a(j, i) = 0) \end{cases} \quad (30)$$

(6) 计算权重

首先由矩阵 R 定义一个有向，带权重的图 $G(A, R)$ ，此图的出边值为：

$$S_L(a, R) = \sum_{C \in A \setminus \{a\}} R(a, c) \quad (31)$$

然后对于所有的 $t_i \in T$ ，可得到相应的权重系数：

$$\lambda(t_i) = \frac{S_L(t_i, R)}{\sum_{t_i \in T} S_L(t_i, R)} \quad (32)$$

然后对目标函数的权重进行规范化。这样，各个目标函数（延误，停车次数，通行能力）的隶属度函数的权重值通过上述方法即可求得。

(7) 确定 p 值

①当 $p = 1$ 时，目标函数表式三个单一目标函数的距离之和，将此目标函数成为 Manhattan 距离[9]，即：

$$L = \max \left[\sum_{i=1}^3 (\lambda_i u_i) \right] \quad (33)$$

②当 $1 < p < +\infty$ 时，目标函数表示三个单一目标函数的加权距离之和，特别地，当 $p = 2$ 时，目标函数表示 Euclidean 距离[10]，即

$$L = \max \sqrt{\sum_{i=1}^3 (\lambda_i u_i)^2} \quad (34)$$

③当 $p = +\infty$ 时，目标函数表示三个单一目标函数的最大加权距离，将此目标函数成为 Chebyshev 距离[11]，即；

$$L = \max \left[\min \sum_{i=1}^3 (\lambda_i u_i) \right] \quad (35)$$

为了得到合适的信号配时参数，需要分别对 p 取上述值进行求解，然后对比所得到的结果，选择合适的信号配时参数。

5.1.4 Webster 算法的模型求解

在这里我们假设各个方向的车道在交叉口右转的时候，不需要看红绿灯。所以我们将题目中所给的车道按照经典的相位划分[12]，其划分的结果如图 1：

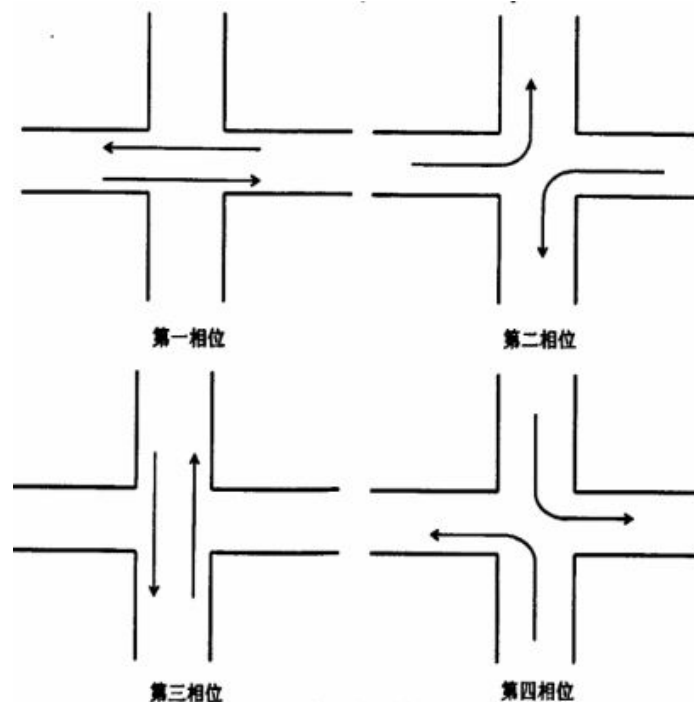


图 1 经典四相位的划分

然后，根据题目所给的数据，在这里我们假设饱和流量值为 4000，由上述的相位分类方法，得到如表 2 和表 3 的通流量和相位饱和度，

表 2：交叉口 A 的各个相位的

	第一相位		第二相位		第三相位		第四相位	
	东直	西直	东左	西左	南直	北直	南左	北左
交通流量	1394	166	366	295	408	394	525	100

饱和流量	4000	4000	4000	4000	4000	4000	4000	4000
相位饱和度	0.349	0.04145	0.0915	0.07375	0.102	0.0985	0.13125	0.025
$\max(y_i^1, y_i^2)$	0.349		0.0915		0.102		0.13125	

表 3: 交叉口 B 的交通流量和饱和流量

	第一相位		第二相位		第三相位		第四相位	
	东直	西直	东左	西左	南直	北直	南左	北左
交通流量	1154	304	802	450	420	535	169	132
饱和流量	4000	4000	4000	4000	4000	4000	4000	4000
相位饱和度	0.2885	0.076	0.2005	0.1125	0.105	0.1385	0.04225	0.033
$\max(y_i^1, y_i^2)$	0.2885		0.2005		0.1385		0.04225	

根据上述公式 (8)，基于单个目标函数最优，假设车辆启动损失时间 3s 我们得到了 A 有效绿灯时间 (59, 16, 17, 22)，B 有效绿灯时间 (49, 32, 23, 8)。

5.1.5 遗传算法的模型求解

遗传算法[13]的基本操作如下：

①编码

把所需要选择的特征进行编号，每一个特征就是一个基因，一个解就是一串基因的组合。为了减少组合数量，在图像中央进行分块，然后再把每一块看成一个基因进行组合优化计算。每个解的基因数量是要通过实验确定的。

②初始群体的生成

随机产生 N 个初始串结构数据，每个串结构数据称为一个个体。 N 个个体，构成了一个群体。GA 以这 N 个串结构数据作为初始点开始迭代。这个参数 N 需要根据问题的规模而确定。

③杂交

由交换概率 (p_c) 挑选的每两个父代通过将相异的部分基因进行交换，从而产

生新的个体。可以得到新一代个体，新的个体组组合了其父辈个体的特性。交换体现了信息交换的思想。

④适应度值评估检测

计算交换产生的新个体的适应度。适应度用来度量种群中个体优劣（符合条件的程度）的指标值。

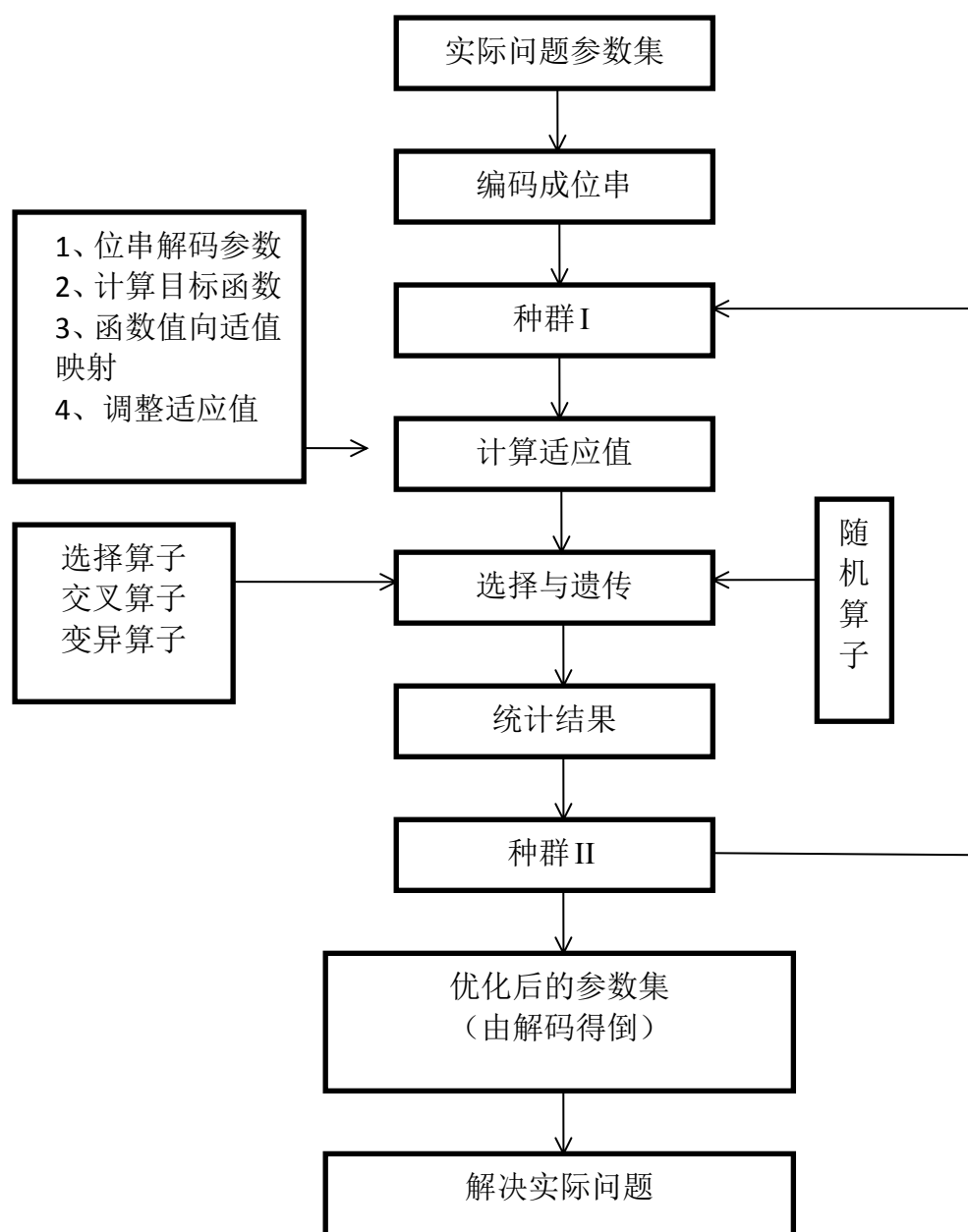
⑤选择

选择的目的是为了从交换后的群体中选择出优良的个体，使他们有机会作为父代为下一代繁衍子孙。遗传算法通过选择过程体现这一思想，进行选择的原则是适应性强的个体为下一代贡献的概率大，选择实现了达尔文的适者生存的原则。

⑥变异

变异首先在群体中随机选择一定数量个体，对于选中的个体以一定的概率随机地改变串结构数据中某个基因的值。

遗传算法的具体运算流程图如图 2：



遗传算法的实现步骤:

- (1) 对所求问题的参数范围进行评估, 并对其进行编码;
- (2) 随机产生由多个个体组成的初始种群;
- (3) 对种群中的所有个体进行解码, 并通过解码后的参数对适应度函数进行求解, 从评估出每个个体的适应度;
- (4) 判断收敛条件, 如果最优解已经得到, 则停止搜索, 否则继续遗传操作;
- (5) 选择, 使得适应度值大的个体越来越多, 适应度值较小的个体会被淘汰;
- (6) 交叉, 让两个个体以某一概率进行交叉操作, 从而产生两个新个体;
- (7) 变异, 让某个个体以一定的概率进行变异操作, 使得个体的特性发生改变;
- (8) 重复步骤 (3) 至 (7), 直到参数收敛为止

根据建立的数学模型, 利用 MATLAB 遗传算法工具箱来求解四相位的单点信号配时优化问题。算法参数设置为: 个体数目 100, 最大遗传代数 20, 变量的二进制位数 20, 代沟 0.9, 交叉概率 0.9。在 MATLAB 下运行程序(见附录 8.1), 得到路口 A 的四相位有效绿灯时间为 53,15,20,23, 路口 B 的有效绿灯时间为 47,20,21,16

遗传算法迭代过程如图 3 所示。

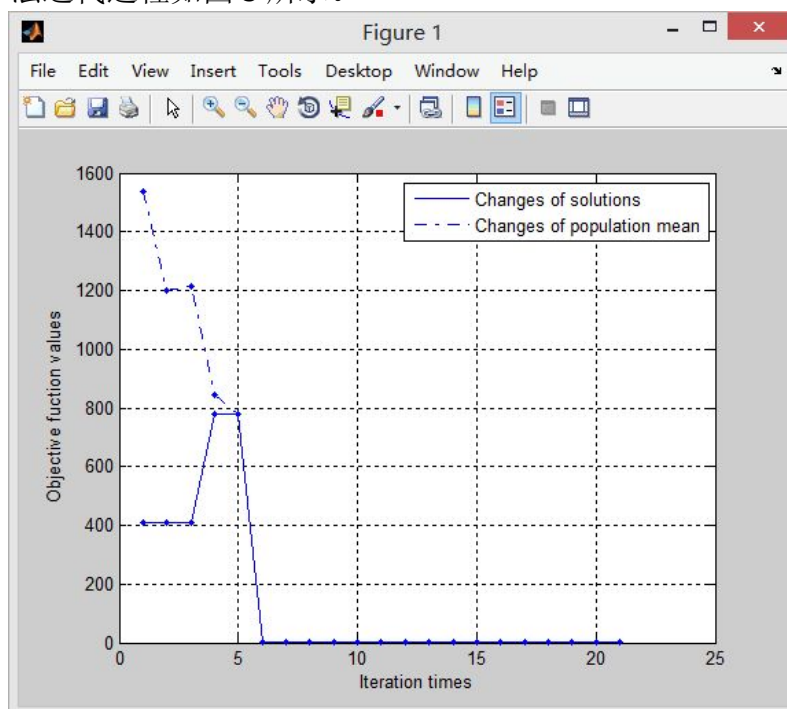


图 3 遗传算法迭代过程

5.1.6 蚁群算法的模型求解

蚁群算法[14]是采用计算机对蚂蚁觅食这一行为作了智能模拟, 该算法是的具体操作流程如图 4:

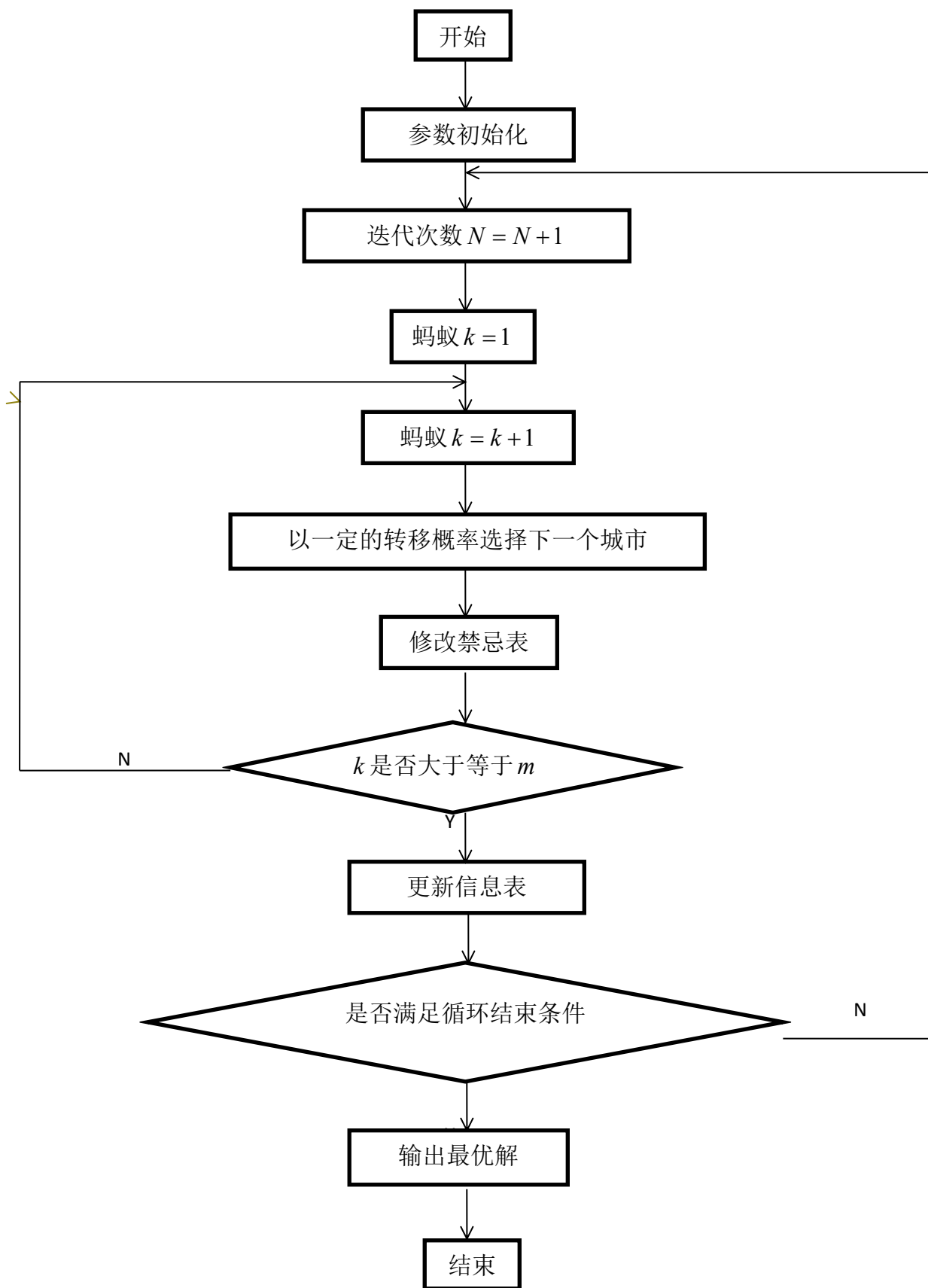


图 4 基于蚁群算法流程图

蚁群算法的实现步骤：

(1) 算法中的所有参数初始化, 即时间 $t=0$, 迭代次数 $N=0$, 设置最大迭代次数 N_{\max} , 将 m 只蚂蚁都置于 n 个城市上, 蚂蚁 $k=1$, 令初始时刻 $\pi_{ij}(0)=C$,

$\Delta\pi_{ij}(0)=0$, 赋予 ρ 和 Q

(2) 迭代次数 $N \leftarrow N+1$;

(3) 蚂蚁数目 $k \leftarrow k+1$;

(4) 每只蚂蚁都按照状态移动概率 $p_{ij}^k(t)$ 去选择城市 j 并行进;

(5) 等蚂蚁选择后将蚂蚁移动到新城市, 并把该城市移到该蚂蚁的禁忌表中;

(6) 若没有走完 n 个城市, 则转 (3), 否则转 (7);

(7) 对每条路线上的信息量进行更新;

(8) 若迭代次数 $N \geq N_{\max}$, 则迭代过程结束, 否则转 (2)。

利用 MATLAB 用蚁群算法来求解四相位的单点信号配时优化问题。算法参数设置为: 蚂蚁数量 50, 迭代次数 500, 蚂蚁爬行速度 0.3, $\rho=0.85$, $Q=0.8$ 。

在 MATLAB 下运行程序(见附录 8.2), 得到路口 A 的四相位有效绿灯时间为 50,12,20,21, 路口 B 的有效绿灯时间为 40,18,29,10。蚁群算法迭代过程如图 5 所示。

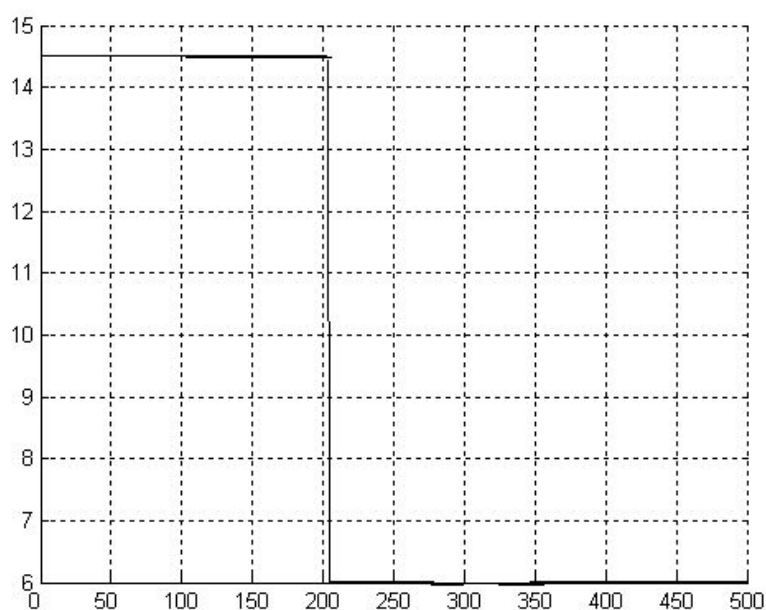


图 5 蚁群算法迭代过程

5.1.7 不同算法的对比分析及最终方案的确定

①时间复杂度[15]

时间复杂度是指执行算法所需要的计算工作量。一般情况下, 算法的基本操作重复执行的次数是模块 n 的某一个函数 $f(n)$, 因此, 算法的时间复杂度记做: $T(n)=O(f(n))$ 。在计算时间复杂度的时候, 先找出算法的基本操作, 然后根据相应的各语句确定它的执行次数, 再找出 $T(n)$ 的同数量级, 找出后, $f(n) =$ 该数量级, 若 $T(n)/f(n)$ 求极限可得到一常数 c , 则时间复杂度 $T(n) = O(f(n))$ 。

为了比较不同算法的时间复杂度, 我们用时间效率作为度量标准, 将算法转

化为程序，在计算机上运行，通过计算机内部的计时功能获得精确的时间，然后进行比较。我们发现蚁群算法程序的运行比遗传算法需要更多的时间，且遗传算法迭代 20 次，蚁群算法迭代 500 次，我们有理由相信，从时间复杂度上看，遗传算法更好。

②空间复杂度[15]

空间复杂度是对一个算法在运行过程中临时占用存储空间大小的量度，记做 $S(n)=O(f(n))$ 。一个算法在计算机存储器上所占用的存储空间，包括存储算法本身所占用的存储空间，算法的输入输出数据所占用的存储空间和算法在运行过程中临时占用的存储空间这三个方面。而存储算法本身所占用的存储空间与算法书写的长短成正比。

在此，我们只比较存储算法本身，也即算法书写的长短。比较发现遗传算法程序较为简单，我们有理由相信，从空间复杂度上看，遗传算法更好。

③收敛性

根据图 3 发现遗传算法收敛速度较快，经六次迭代后就得到问题的最优解，但该算法稳定性较差，目标函数值波动较大。但由图 4 发现蚁群算法比遗传算法收敛速度更快，稳定性更好。

④总延误，通行能力，停车次数的比较

根据公式，计算 Webster 算法，遗传算法，蚁群算法的总延误，总停车次数及总通行能力，结果见表 4 和表 5：

表 4 A 交叉路口的算法的实验结果

A 交叉口	有效绿灯时间 (g_1, g_2, g_3, g_4)	周期长度	总延误	总停车次数	总通行能力
改善前	(52,19,31,22)	140	3429374	1227913	7086
Webster 算法	59, 16, 17, 22	130	5962354	2796354	6534
遗传算法	53,15,20,23	125	2463218	1003765	8647
蚁群算法	50,12,20,21	119	1732596	1739648	8973

表 5 B 交叉路口的算法的实验结果

B 交叉口	有效绿灯时间 (g_1, g_2, g_3, g_4)	周期长度	总延误	总停车次数	总通行能力
改善前	(43,17,35,18)	129	4852495	462309	7008
Webster 算法	49, 32, 23, 8	124	7324590	587469	5987
遗传算法	47,20,21,16	110	3782965	376582	8092
蚁群算法	40,18,29,10	97	2732549	436975	8913

分析可得 Webster 算法求得的周期长度、总延误时长和总停车次数数值较大，且总通行能力较弱。与 Webster 算法相比，用遗传算法和蚁群算法求得的周期长度有所降低，总延误和总停车次数下降两倍多，有了显著减少，并且总通行能力也有了明显增强。与遗传算法相较之，用蚁群算法求得的周期长度和总延误都有所降低，总通行能力有了一定提高，但是总的停车次数较高。对于四相位的信号配时优化问题，蚁群算法的综合性能较之 Webster 算法和遗传算法更好，可

以很好地满足实时交通需求。

根据上述分析的所有指标，我们得到最终的单路口的配时优化方案为：对于 A 交叉路口各相位的配时优化方案为 54,16,24,25，B 交叉路口各相位的配时优化方案为 44,22,34,14。

5.2 相邻交叉口相位差优化模型的建立与求解

由于在本题中，给出了两个相邻的交叉口，所以我们需要建立两个相邻交叉路口的相位差的优化模型，我们认为该问题分为以下几个步骤：

步骤一：建立相邻交叉口相位差优化模型，分别对问题一的车辆延误、平均停车次数、交叉口的通行能力模型进行修改。

步骤二：在修改后的模型基础上，进行计算求解出相邻交叉口相位差。

5.2.1 相位差和周期

用 t 来表示相邻两交叉口绿灯的相位差，则有

$$t_{i,j} = l/v$$

其中， l 是相邻两交叉口 i 和 j 之间的距离， v 为平均车速。

为了使双向行驶的车辆都满足上式，相邻两交叉口必须采用相同的信号周期，即

$$C_i = C_j$$

令 $C_i = C_j = C$ ，则有

$$t_{i,j} + t_{j,i} = nC \quad (36)$$

其中， n 为正整数。当 $n=1$ 时，有

$$t_{i,j} + t_{j,i} = C \quad (37)$$

5.2.2 车辆延误模型

交通信号控制的主要目标是获得最小的车辆延误，它是最常用的一个性能指标。因此，在上述公式的基础上，交叉口 j 第 i 相位的车辆延误可定义为：

$$D_{i,j} = \frac{Cq_{i,j}(x - y_{i,j})^2}{2x^2(1 - y_{i,j})} + \frac{x^2}{2(1 - x)}, \quad i, j = 1, 2, \dots, n \quad (38)$$

其中， C 为信号周期长度 (s)， $q_{i,j}$ 为交叉口 j 第 i 相位的交通流量 (pcu/h)，

x 为交叉口 j 的饱和度， $y_{i,j}$ 为交叉口 j 第 i 相位的饱和度。

5.2.3 平均停车次数

在上述公式的基础上，交叉口 j 第 i 相位的平均停车次数可定义为：

$$H_{i,j} = \frac{y_{i,j}(x - y_{i,j})}{2x^2(y_{i,j} - x^2)}, \quad i, j = 1, 2, \dots, n \quad (39)$$

5.2.4 交叉口通行能力

基于公式上述公式，交叉口 j 第 i 相位的交叉口通行能力可定义为：

$$Q_{i,j} = s_{i,j} \cdot \left[1 - \frac{y_{i,j}(x - y_{i,j})(1 - y_{i,j})}{1.8x^2(y_{i,j} - x^2)} \right], \quad i, j = 1, 2, \dots, n \quad (40)$$

其中， $s_{i,j}$ 为交叉口 j 第 i 相位的饱和流量（pcu/h）。

5.2.5 优化模型及其求解

本论文将车辆延误最小化、平均停车次数最小化和交叉口通行能力最大化作为主干路信号配时优化的三个重要目标，并且采用统一目标函数法将主干路信号配时优化问题转化为最小化问题，对应的目标函数则可以通过三个权系数转化为包含两个自变量 C 和 x 的单目标函数，其中权系数随交通需要的变化而变化。为了简化主干路信号配时优化模型，将目标函数表示为：

$$\min f(x, C) = \sum_{j=1}^n \sum_{i=1}^n (K_{i,j}^1 D_{i,j} + K_{i,j}^2 H_{i,j} - K_{i,j}^3 Q_{i,j}) \quad (41)$$

约束条件为：

$$\begin{cases} C_i = C_j = C \\ t_{i,j} + t_{j,i} = nC \\ C \in [40, 120] \\ x \in [0.6, 0.8] \end{cases} \quad (42)$$

其中， $K_{i,j}^1$ 、 $K_{i,j}^2$ 和 $K_{i,j}^3$ 分别是车辆延误，平均停车次数和交叉口通行能力的权系数。

根据城市交通的特征，权系数 $K_{i,j}^1$ 、 $K_{i,j}^2$ 和 $K_{i,j}^3$ 的定义式可由以下三个方面得到。

(1) 要求 $K_{i,j}^1$ 和 $K_{i,j}^2$ 随着饱和度 $y_{i,j}$ 的增大而减小，而 $K_{i,j}^3$ 随着饱和度 $y_{i,j}$ 的增大而增大，从而增强高峰期的交叉口通行能力，减少平峰期的车辆延误和平均停车次数；

(2) 要求 $K_{i,j}^1$ 随着饱和流量 $s_{i,j}$ 的增大而增大。从而减少大流量交叉口处的车辆延误；

(3) 一般地，周期时间越长，交叉口通行能力越强，所以在 $K_{i,j}^3$ 的定义式中引入周期长度 C ，从而增强交叉口通行能力。

因此权系数 $K_{i,j}^1$ 、 $K_{i,j}^2$ 和 $K_{i,j}^3$ 的定义如下：

$$\begin{cases} K_{i,j}^1 = 2 \cdot (1-Y) \cdot \sqrt[7]{s_{i,j}} \\ K_{i,j}^2 = \sqrt[7]{s_{i,j}} \cdot \frac{1-Y}{0.9} \\ K_{i,j}^3 = 2Y \cdot \frac{C}{3600} \end{cases} \quad (43)$$

其中, $Y = \sum_{j=1}^n \sum_{i=1}^n \max(y_{i,j}^1, y_{i,j}^2, \dots, y_{i,j}^r)$, $y_{i,j}^r$ 是交叉口 j 第 i 相位 r 方向上的饱

和度。

在求解的过程中。我们假设 A、B 两个路口之间的距离为 500m, 车流的速度为 36km/h , 通过计算两个相邻交叉口的相位差为 9s 。

5.3 仿真检验

基于 Vissim 仿真平台[16], 建立武汉市交叉路口的微观仿真模型, 模拟其交通运行情况, 并且运用上文提出的基于交叉口的总延误、总停车次数、总通行能力在交叉口配时方面相互联系和制约, 在鲁棒性能上相对于其他模型一定程度上的提高, 加强了配时方案的稳定性。

交通的仿真的步骤如图 6:

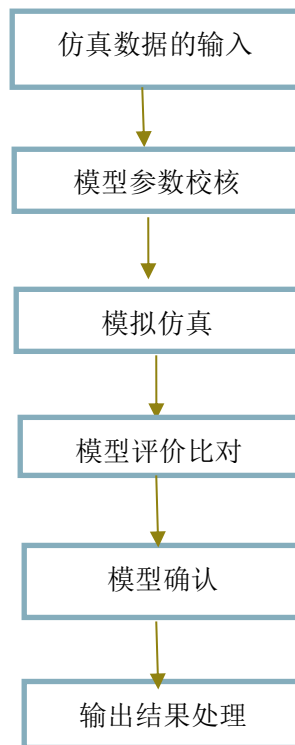


图 6 交通仿真流程图

利用 Vissim 建立网络主要用到两个组件, 即路段(Link)和连接器(Connector)。用一个 Link 表示一条车道还是多条车道, 以及 Connector 的连接等问题要根据具体情况而定。

(1) 进口到 Link 的设置

由于在本题中没有特殊的说明设置有专用的车道, 这里假设车道的无功能划分, 可以用一个进行车道划分后的 Link 来表示。

(2) 交叉口的进口道和出口道的链接

交叉口的进口道和出口道是通过 **Connector** 连接在一起的，在每一个进口都有三条车道，分别用三个 **Link** 来表示，因为不同的 **Link** 之间的车道是不同换道的，所以车辆只能在 **Link** 内进行换道行驶。完成上述的两个步骤之后得到如下交叉口路网建模结果如图 7 所示。

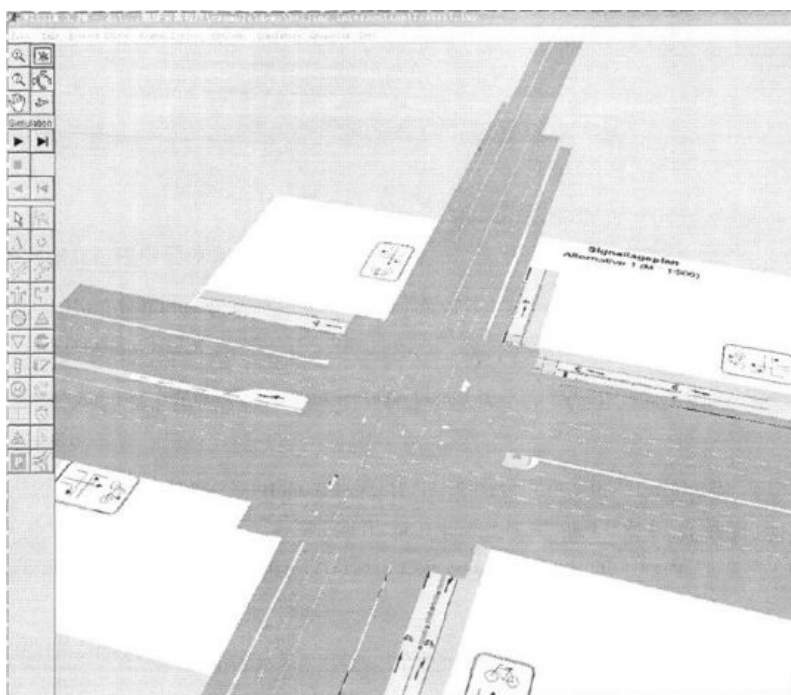


图 7 十字交叉口的建模示意图

在单独使用 **VISSIM** 软件进行仿真时，交通流量的输入和路径的选择都在 **VISSIM** 中提前配置好；在使用实时数据仿真系统进行仿真时，由于车辆由系统控制按实际的时间生成车辆，因此不用输入交通量。仿真时间为 600s。其仿真结果如图 8。



图 8 VISSIM 仿真图

下面我们给出三次仿真结果的值见表 6：

表 6 :VISSIM 仿真结果评价表

参数	车均延误时间	平均停车时间	停车次数	通过车辆数
bx_mult=3.25	32.5	27.8	2.17	997
bx_mult=3.5	31.7	26.6	2.12	997
bx_mult=3.75	31.0	25.7	2.08	998
bx_mult=4	33.9	29.5	2.23	997

由表 6 的数据可以看出，单独使用 VISSIM 仿真软件进行仿真时，车辆按经验分布（泊松分布）得到车辆生成时间，车辆生成的时间并不确定，因此随着参数 bx_mult 的增大，车均延误时间没有规律性的变化而且波动性很大。

六、模型评价

6.1 模型优点

为优化相邻交叉口信号配时问题，本文分别对四相位的单点信号配时优化和相邻交叉口相位差优化问题建立数学模型。本文在以下几方面有所创新，首先，在信号配时中，我们选取多个目标进行最优化。同前人多数研究来看，大都只是把延误时间作为唯一影响优化目标的因素，这样就是造成考虑问题过于片面，使得结果与真实情况有较大偏颇，而且基于一点影响因素所得到的目标函数并不是最优的。综合多方面的考量，我们从停车次数、平均停车次数、交叉口的通行能力等多方面的设计规划影响车辆延误的因素。另外，在单点信号配时问题模型的建立与求解中，我们在 Webster 算法基础上，运用模糊折中多目标规划，基于蚁群算法和遗传算法求解，多种算法相互结合使模型更具科学性，逻辑严密性，同时也弥补了一个算法求解的不足。最后，我们求解出改善后的交通信号配时方案并进行仿真检验，增强模型的说理力，以及方案的可行性。

6.2 模型缺点

此模型的建立还有一些不足，由于数据的冗杂性和时效性，导致我们对于数据的搜集并不十分完整，使得我们得到的结果与预期相比有一定的出入。另外，此模型假设不考虑非机动车道，人行道和特殊人士准用通道，但在实际生活中，非机动车和行人对交通指挥存在一定影响，特别是在一些大中型城市中，这一问题尤其突出，而且专用特殊车道的设置也极大地影响了车流的速度。

6.3 模型推广

由于，数学模型的建立，其目的时在于能够应用在实际的生活中，优化生活中的某些问题，如同本题中的相邻两交叉路口的配时优化问题，在模型建立的过程中，我们并没有考虑非机动车辆和行人对相邻两交叉路口的配时的影响，而且对于不同的路段，还有特殊的专用车道，比如公交车专用车道，应急车道等等，然而，在实际的生活中，非机动车辆和行人对相邻两交叉路口的车辆延误时间、平均停车次数、交叉口的通行能力影响都很大，以至于不能够忽略，所以可以考虑在三个主要影响因子的模型上，加上非机动车辆和行人这个影响因子对车辆延误时间、平均停车次数、交叉口的通行能力影响，通过实地调查，确定该影响因子的大小。

另外，由于车流量并不是一直恒定的，在不同的时间段其差别是很大的，例如在上下班期间，其车流量是深夜的好几倍，因此，可以通过分时段的方法，来调节红绿灯的配时，这样会提高整个道路的通行能力。

七、参考文献

[1] 赵雨旸, 冯雨芹, 杨忠良, 信号交叉口 Webster 法延误计算修正模型, 2010,

6 (24) , 2-3。

[2]陈绍宽, 郭谨一, 王璇, 毛保华, 信号交叉口的延误计算方法的比较, 2005, 6 (29) , 77-78

[3]王殿海, 祁宏生, 徐程, 陈松, 信号交叉口停车次数, 2009, 9 (39) , 141-144

[4]李志平, 城市信号控制交叉口通行能力研究, 2010, 4, 37-39

[5]许焕卫, 黄洪钟, 张旭, 基于模糊折中规划的稳健多目标优化设计, 2007, 5 (47) , 368-369

[6] <http://baike.haosou.com/doc/6929460-7151669.html>, 2015-5-17

[7]http://baike.baidu.com/link?url=-7wmYcBdAikAs0uXkZnxDS4r2nmJlG6DoY9OvnmNmp41dZuNaVSeeVZr7Ik5d0zWzWtvOK5846q_PJZoLXMfWK, 2015-5-18

[8]关志华, 寇纪淞, 黎民强, 基于模糊偏好的多目标进化优化算法, 2002, 35(3), 275-279.

[9].http://baike.baidu.com/link?url=uYq1BxUXgMppHg41aB9yjW11qBjs9kX_k2xjBDdBHbVZNln_QQ1E6KawC5sStUSiA, 2015-5-18

[10]http://baike.baidu.com/link?url=r150K28tnDAaGO8KLaF3rSnhkdG402gbsH_F7xBksVPdnM2443nJbp246sFwIZO25dbTF2LOzGHidcm-7zJVka, 2015-5-18

[11] <http://blog.csdn.net/cscj2010/article/details/7791640>, 2015-5-18

[12]刘瑞婷, 城市平面正交交叉口信号配时模型研究, 2008,34-35

[13] <http://www.docin.com/p-111086110.html>, 2015-5-17

[14] <http://baike.haosou.com/doc/5430490-5668760.html>, 2015-5-17

[15]http://baike.baidu.com/link?url=wZQSEnOcV43NUJWlk5y44U8z6Xjw6QqY8Ud4nxxRP4SSmdYCqvhk2u2dAzKDaBFDM_VtaQp16qN_-xmsUJBq8K#3

[16]<http://www.doc88.com/p-8059993564994.html>, 2015-5-18

八、附录

8.1 遗传算法的程序

注意: 其中在使用 MATLAB 调试时要添加遗传算法工具箱

NIND=100;

MAXGEN=20;

NVAR=2;

PRECI=20;

GGAP=0.9;

trace=ones(MAXGEN+1,2);

FieldD=[repmat([PRECI],[1,NVAR]);[40,0.6;120,0.8]; repmat([1;0;1;1],[1,NVAR])];

Chrom=crtbp(NIND,NVAR*PRECI);

v=bs2rv(Chrom,FieldD);

gen=1;

kk=100;

while kk>=4

 [NIND,N]=size(Chorm);

 M=fix(NIND/2);

 ObivV=f(v(1:M,:));



```

    FitnV=ranking(ObjvV);
    SelCh=select('sus',Chorm(1:M,:),FithV,GGAP);
    SelCh=recombin('xovsp',SelCh,0.9);
    Chrom=mut(SelCh);
    kk=size(Chorm)
    v=bs2rv(Chrom,FieldD);
    trace(gen,1)=min(f(v));
    trace(gen,2)=sum(f(v))/length(f(v));
    gen=gen+1;
end
figure();clf;
plot(trace(:,1));hold on;plot(trace(:,2),'-');
plot(trace(:,1),'');plot(trace(:,2),'');grid;
legend('Changes of solutions','Changes of population mean')
xlabel('Iteration times');ylabel('Objective function values')
8. 2 蚁群算法的程序
function [BESTX,BESTY,ALLX,ALLY]=ACOUCP(K,N,Rho,Q,Lambda,LB,UB)
K=20
N=20
Rho=0.85
Q=0.80
Lambda=0.3
LB=[40,0.6]
UB=[120,0.8]
M=length(LB);
X=zeros(M,N);
for i=1:M
    x=unifrnd(LB(i),UB(i),1,N);
    X(i,:)=x;
end
ALLY=cell(K,1);
ALLY=zeros(K,N);
BESTX=cell(K,1);
BESTY=ones(K,1);
k=1;
Tau=zeros(1,N);
Y=zeros(1,N);
while k<=K
    YY=zeros(1,N);
    for n=1:N
        x=X(:,n);
        YY(n)=fit(x,Y);
    end
    maxYY=min(YY);

```

```

temppos=find(YY==maxYY);
POS=temppos(1);
for n=1:N
    if n~=POS
        x=X(:,n);
        Fx=FIT(x)
        mx=GaussMutation(x,LB,UB)
        Fmx=FIT(mx)
        if Fmx<Fx
            X(:,n)=mx;
            Y(n)=Fmx;
        elseif rand>1-(1/(sqrt(k)))
            X(:,n)=mx;
            Y(n)=Fmx;
        else
            X(:,n)=x
            Y(n)=Fx;
        end
    end
end
for n=1:N
    if n~=POS
        x=X(:,n);
        Fx=FIT(x);
        mx=GaussMutation(x,LB,UB);
        Fmx=FIT(mx);
        if Fmx<Fx
            Y(n)=Fmx;
        else if rand>1-(1/(sqrt(k)))
            X(:,n)=mx;
            Y(n)=Fmx;
        else
            X(:,n)=x;
            Y(n)=Fx;
        end
    end
end
for n=1:N
    if n~=POS
        x=X(:,n);
        r=(K+k)/(K+K);
        p=randperm(N);
        t=ceil(r*N)
        pos=p(1:t)
    end
end

```

```

        TempTau=Tau(pos)
        maxTempTau=max(TempTau)
        pos2=find(TempTau==maxTempTau)
        pos3=pos(pos2(1))
        x2=X(:,pos3(1))
        x3=(1-Lambda)*x+Lambda*x2
        Fx=FIT(x);
        Fx3=FIT(mx);
        if Fx3<Fx
            X(:,n)=x3;
            Y(n)=Fx3;
        else if rand>1-(1/(sqrt(k)))
            X(:,n)=x;
            Y(n)=Fx;
        end
    end
end
Tau=Tau*(1-Rho)
maxY=max(Y)
minY=minY
DeltaTau=(maxY-Y)/(maxY-minY);
Tau=Tau+Q*DeltaTau
ALLX{k}=X;
ALLY(k,:)=Y;
pos4=find(Y==minY);
BESTX{k}=X(:,pos4(1));
BESTY(k)=minY;
disp(k);
k=k+1;
end
end
end

plot(BESTY,'-ko','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize','2')
ylabel('Objective function values')
xlabel('Iteration times')
grid on

```