

全国第五届研究生数学建模竞赛



题目 城市道路交通信号实时控制问题

摘 要

本文通过分析实际道路交通状况，在合理假设的基础上，建立了孤立十字、丁字交叉路口、三个交叉路口线状区域以及六个交叉路口网络状区域的交通信号实时配时数学模型，利用 Matlab 软件产生满足泊松分布的车流量实时序列，按照优化目标进行了算法的设计和编程，得到了相应的数值结果，并对结果进行了讨论。

首先我们对单个交叉路口的车辆总平均等待时间建立了数学模型，以总延误时间最小为优化目标给出实时算法，与韦伯斯特算法进行比较后优势明显。

其次建立了包括线状和网络区域在内的多个交叉路口实时配时模型，采用二次优化算法，与传统交通配时算法相比，等待时间分别缩短了 9.5%和 11.3%。

最后，在以按泊松分布而产生的车流量序列为输入的情况下，使用上述点控制，线控制，面控制算法进行了交通信号实时配时，达到了总平均等待时间最小的目标。并在点控制算法中，比较了变周期变绿信比，定周期变绿信比，定周期定绿信比三种算法，得出了在实时流量序列下变周期变绿信比效果最佳的结论。

关键词：交通信号 实时配时 优化算法 泊松过程

参赛队号 1024710

参赛密码 _____
(由组委会填写)

一、 问题重述

城市交通管理的基本问题之一，是城市道路交通信号控制问题。即根据不同交叉路口、不同相位、不同方向、不同时段交通流量（标准车辆数），合理地配置各交叉路口的信号灯的周期长，以及同一周期内的红、绿、黄信号的响应时间。以前大多采用固定周期，固定信号比的配时控制方案。为提高道路服务功能，设计实时优化的配时方案对道路畅通和应急决策管理具有重要意义。设计实时配时方案，即根据交通流量的实时大小，实时配置信号灯的周期长、各种色灯的响应时间，同时考虑信号灯的转换与车辆的起动的损失时间，使全体车辆在所有道口的等待的时间最短。（1）构造单个交叉路口(十字路口或丁字路口)交通信号实时控制的点控制数学模型，并给出相应的实时算法。（2）构造多个交叉路口(线状区域)交通信号实时控制的线控制（至少 2 个交叉口）数学模型，并给出相应的实时算法。（3）构造多个交叉路口(网络区域)交通信号实时控制的面控制（至少 5 个交叉口）数学模型，并设计相应的实时算法。（4）根据城市交通流分布规律(一般理解为 Poisson 分布)，设计一种实时产生交通流序列的方案。并根据你的算法和产生的交通流数据，计算并给出单交叉路口点控制的实时信号配时方案(分为周期固定和周期不固定两种情形考虑)，并与固定配时方案比较,说明实时配时方案的效果和优势。（5）对多路口信号配时的情况，根据你产生的交通流数据和相应的实时算法，分别给出线状区域、网络区域实时配时方案，并比较和评价你所得到的结果。同时分析模型算法的可计算性、算法的复杂性。（6）给交通管理部门提出应用你所得结果的咨询和建议（例如：流量预测方法、数据处理方法，软件设计、实现步骤等）。

二、 问题分析

城市道路系统由路段和交叉口组成，由于不同方向的交通流在交叉口处相交，引起交通流之间的冲突、合流、分流等交通行为。城市交通信号控制就是通过对交叉路口的交通流进行警告、诱导和调节，减少或完全消除可能引发交通事故的交通冲突点，使车辆和行人的延误时间减少、交叉路口相连各车道的通行能力增加，实现交通流的安全性、快速性与舒适性^[1]。

2.1 主要术语

- 1) 交叉口：交通流的汇集与分散点。平面交叉口可以分为三路交叉、四路交叉和多路交叉等。交叉口的形状又可以分为丁字、十字、Y型和环型等，本文主要以丁字和十字型交叉口为主进行论述。
- 2) 周期：信号灯各种灯色轮流显示一次所需的时间，即各种灯色显示时间之总和；或是某主要相位的绿灯启亮开始到下次该绿灯再次启亮之间的一段时间，单位为秒(s)，如图 2.1 所示。

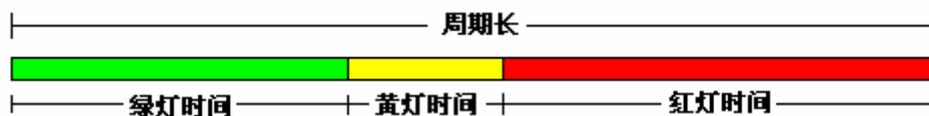


图 2.1 周期长示意

- 3) 信号相位：指一个路口在一个周期时间内按需求人为设定的，同时取得通行权的一个或几个交通流的序列组。
- 4) 绿信比：指在一个周期内交叉路口的各个相位不同方向中某个方向出现绿灯的时间与周期长之比。

- 5) 到达率：交叉路口非饱和情况下到达率为进口道的流量。
- 6) 饱和流量：衡量一个路口交通流施放能力的重要参数，通常是指一个绿灯时间内的连续通过路口的最大车流量（单位：车/秒）。
- 7) 流量系数（流量比）：实际流量与饱和流量的比值。既是计算信号配时的重要参数，又是衡量路口阻塞程度的一个尺度。
- 8) 进口道通行能力：等于进口道饱和流量与该流向所在相位的绿信比之积。
- 9) 交叉口的饱和度：流量与进口道通行能力之比。
- 10) 有效绿灯时间：指被有效利用的实际车辆通行时间。它等于绿灯时间与黄灯时间之和减去头车启动（因为红灯时头车速为零）的损失时间。
- 11) 有效绿灯时间：指被有效利用的实际车辆通行时间。它等于绿灯时间与黄灯时间之和减去损失时间。
- 12) 公路通行能力：指在给定的道路和交通条件下，公路上的某个最小或最困难的断面或某个规定路段上单位时间内平均能够通过的最大车辆数，一般采用小时为单位，故通行能力一般以每小时能够通过的最大车辆数计。

2.2 控制类型

从空间关系上，可分为单路口控制（也称点控）、干线控制（也称线控）、区域控制（也称面控）。

1) 点控制

对城市平面交叉路口或高速公路匝道口独立进行控制，不考虑与相邻交叉路口或匝道口的协调关系，其控制目的是尽可能减少该交叉路口或匝道口的行车延误。

2) 线控制

对含多个平面交叉路口的城市交通干线进行信号控制，其各路口的控制方案相互协调，使进入干线的车队按某一车速行驶时，能不遇或少遇红灯而通过该干线。

3) 面控制

对城市中某个区域的多个平面交叉口进行信号控制，其控制方案相互协调，使得在该区域内某种指标，如总的停车次数、旅行时间、耗油量等最少。

三、符号说明

符号	说明	单位
d_i	单叉路口一周期内第 i 相位每辆车的平均延迟时间	秒/辆
d_{ij}	j 叉路口一周期内第 i 相位每辆车的平均延迟时间	秒/辆
D_i	j 叉路口单位时间内所有车辆的延迟时间	/
l_j	各个叉路口间的距离	米
L_j	j 叉路口一周期的总损失时间	秒
$phase_j$	j 叉路口的相位差	秒
q_i	单叉路口第 i 相位的流量	辆/小时

q_{ij}	j 叉口第 i 相位的流量	辆/小时
s_i	单叉口第 i 相位的饱和流量	辆/小时
s_{ij}	j 叉口第 i 相位的饱和流量	辆/小时
T	单叉口的周期	秒
T_{gi}	单叉口第 i 相位的有效绿灯时间	秒
T_{gij}	j 叉口第 i 相位的有效绿灯时间	秒
v	车辆的平均行驶速度	米/秒
x_i	单叉口第 i 相位的饱和度	/
x_{ij}	j 叉口第 i 相位的饱和度	/

四、模型建立与求解

4.1 基本假设

- 1) 任何一个交叉路口，或称之为叉路口，或为叉口，在空间上是由 4 个路口构成，在时间上是由 4 个相位构成（丁字均为三个）。
- 2) 最大周期长：过长的周期除了增大延误外，从驾驶员、行人等待信号的心理效果来考虑也不好，同时也存在左转交通处理上的问题。实际中最大周期一般取 120s 左右为好，也可以比 120s 更长，但最好不超过 200s。
- 3) 车用最小绿灯时间：为了保证安全，各相位的绿灯时间规定如下：包括直行交通在内的主流交通为 15s 以上，左转专用相位等次交通为 5s 以上，但是对于交通量非常小的交通原则上取 8s 以上。本文取最小绿灯时间为 10s。
- 4) 绿灯损失时间：包括路口绿灯信号开始时前排车辆启动延误时间，及绿灯结束后黄灯时的车辆减速停车损失时间。本文绿灯损失时间取 5s，详细分析见式 4.8。
- 5) 饱和度取值：考虑交通实际情况与本文对道路交通状况所作的假设，饱和度取值为 0.7 至 0.9。
- 6) 其他参数假设：交叉口长度约 20m，车流的平均速度为 10m/s

4.2 单交叉口实时控制模型

4.2.1 基本参数

1) 单交叉十字路口相位

考虑三车道的单交叉十字路口，其四种相位如图 4.1 所示。

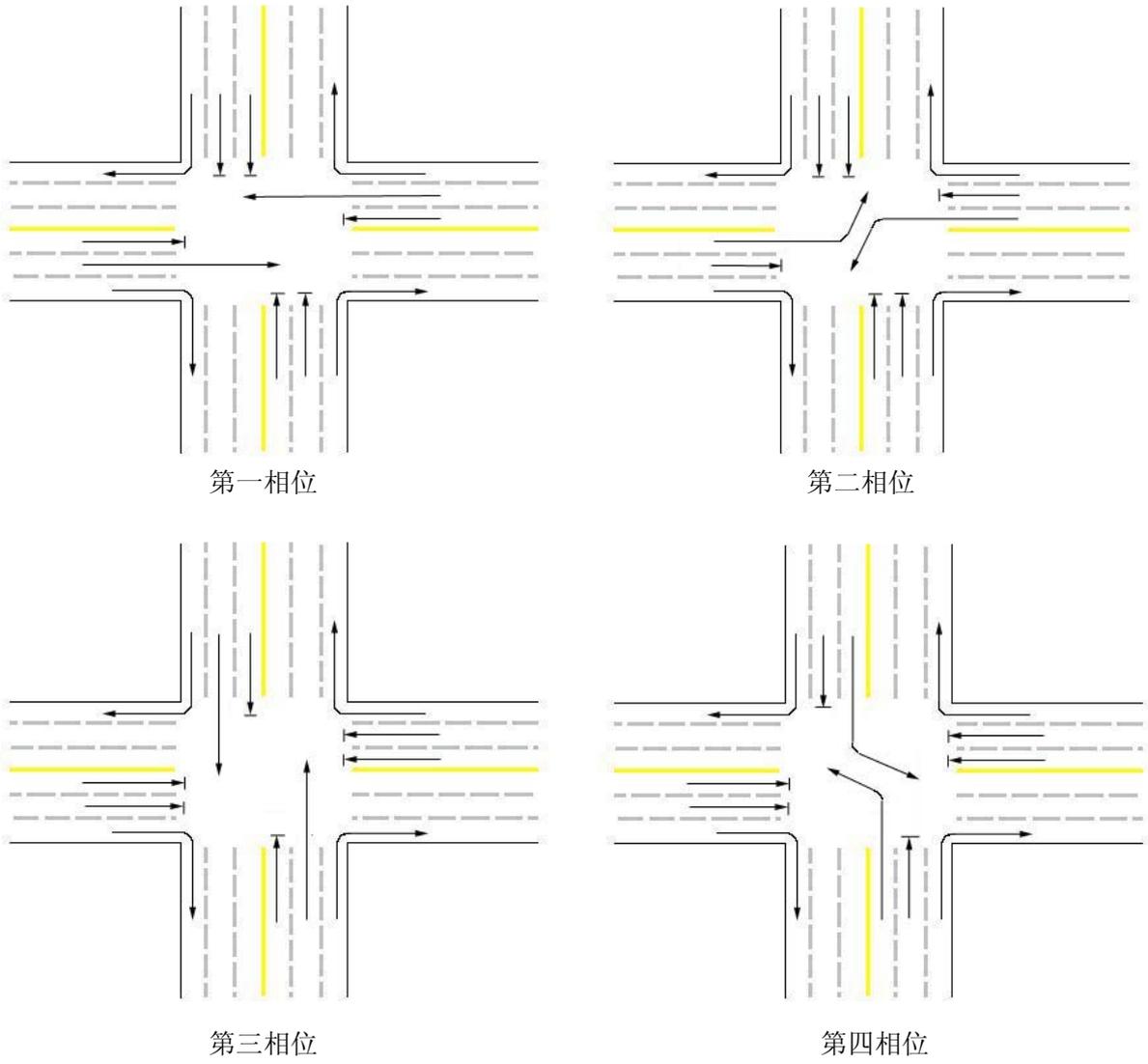


图 4.1 单交叉十字路口相位

2) 单交叉丁字路口相位

考虑两车道的单交叉丁字路口，其三种相位如图 4.2 所示。

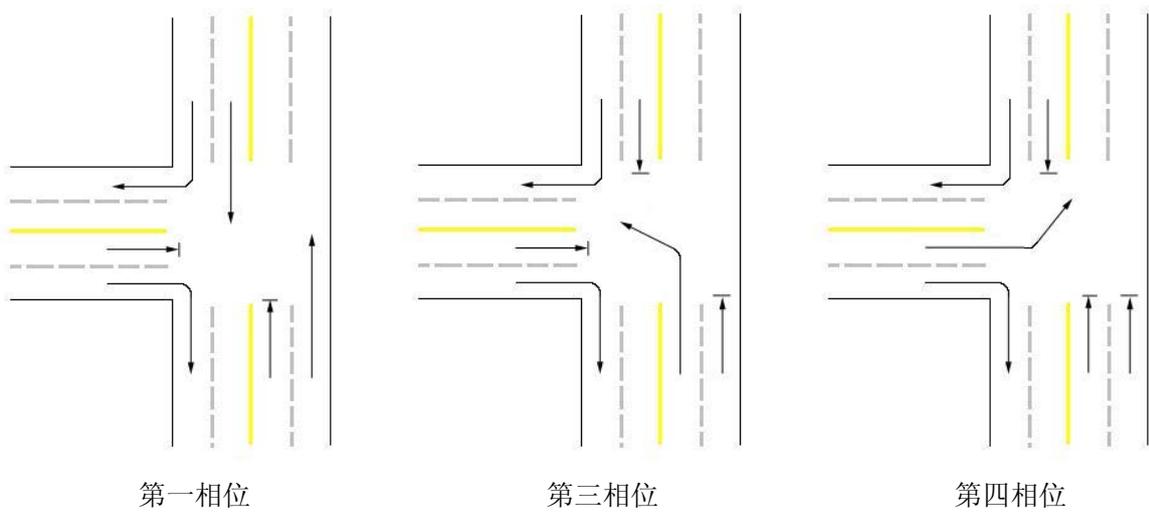


图 4.2 单交叉丁字路口相位

4.2.2 模型建立

1) 基于随机过程的单相位车辆平均等待时间——不考虑车辆排队情况

由于在某一时间段内，到达路口的车辆数量满足泊松分布，所以我们将整个过程视为泊松过程，那么随机过程理论易知相邻两辆车到达的时间差是满足负指数分布的。

A、定理 4.1^[2]

设 $\{N(t), t \geq 0\}$ 为泊松过程，则在已给 $N(t) = n$ 时事件相继发生的时间 S_1, S_2, \dots, S_n 的条件概率密度为

$$f(t_1, t_2, \dots, t_n) = \begin{cases} \frac{n!}{t^n}, & 0 < t_1 < t_2 < \dots < t_n \leq t \\ 0, & \text{其他} \end{cases} \quad (4.1)$$

本定理说明，在 $N(t) = n$ 的条件下， S_1, S_2, \dots, S_n 的条件分布函数与 n 个在 $[0, t]$ 上相互独立同均匀分布的顺序统计量的分布函数相同。

B、单相位车辆平均等待时间

现对单个相位内所有车辆的平均总等待时间进行建模，设到达路口的车辆数量遵照参数为 λ 的泊松分布， $N(t), t \geq 0$ 表示 t 时刻已等候在路口的车辆数量，红灯时间为 t ，

第 i 辆车到达路口的时刻为 S_i ，则 $[0, t]$ 到达路口的车辆，等待时间总和为

$$S(t) = \sum_{i=1}^{N(t)} (t - S_i) \quad (4.2)$$

因为

$$\begin{aligned} E\{S(t) | N(t) = n\} &= E\left\{\sum_{i=1}^{N(t)} (t - S_i) | N(t) = n\right\} \\ &= E\left\{\sum_{i=1}^n (t - S_i) | N(t) = n\right\} = nt - E\left\{\sum_{i=1}^n S_i | N(t) = n\right\} \end{aligned} \quad (4.3)$$

记 $\{Y_i, 1 \leq i \leq n\}$ 为 $[0, t]$ 上独立同均匀分布的随机变量，则由定理 4.1 可知，

$$E\left\{\sum_{i=1}^n S_i | N(t) = n\right\} = E\left(\sum_{i=1}^n Y_{(i)}\right) = E\left(\sum_{i=1}^n Y_i\right) = \frac{nt}{2} \quad (4.4)$$

故

$$E\left\{\sum_{i=1}^n S_i | N(t) = n\right\} = \frac{nt}{2} \quad (4.5)$$

所以

$$\begin{aligned}
E\{S(t)\} &= \sum_{n=0}^{\infty} (P\{N(t) = n\} E\{\sum_{i=1}^{N(t)} (t - S_i) | N(t) = n\}) \\
&= \sum_{n=0}^{\infty} (P\{N(t) = n\} \frac{nt}{2} = \frac{t}{2} E(N(t)) = \frac{\lambda}{2} t^2)
\end{aligned} \tag{4.6}$$

其中， λ 可理解为路口流量， t 为红灯的等待时间，然而在路口等待的车是排成长队的，所以对于任意一辆车，它的等待时间必须加上前面车辆的排空时间，而此模型并未考虑该问题，所以对模型修改如下。

2) 单相位车辆平均等待时间——考虑车辆排队情况

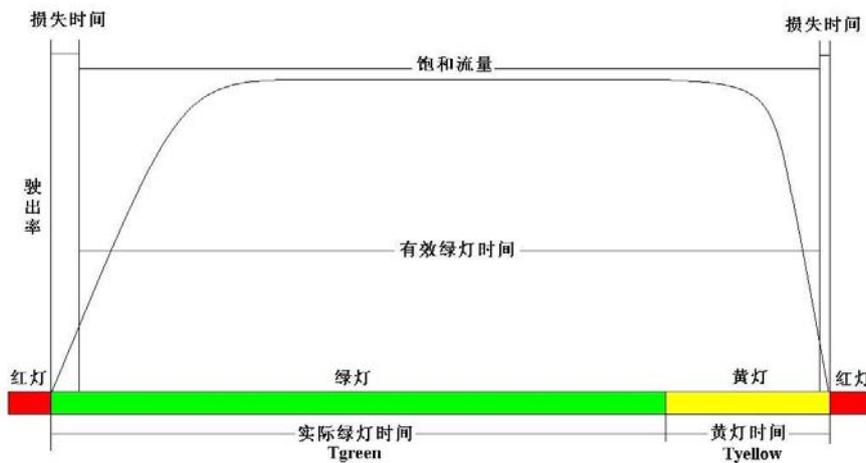


图 4.3 部分控制参数之间的关系

由图 4.3 可以可知，在一个信号相位上，绿灯时间和黄灯时间之和 $T_{green} + T_{yellow}$ 为车辆的可通行时间。但是可通行时间并不能完全得到充分利用：在绿灯信号开始出现的最初几秒，车流量是一个从零上升至饱和流量的过程，这是由于排队的车流正处于起步和加速阶段，造成一定的时间损失，记为 $loss1$ 。同样，在绿灯结束后的黄灯期间，由于部分车辆已采取制动措施，停止前进了，部分车辆虽未停驶，但已经减速，因此这段时间里车辆流量由大变小，逐渐下降至零，黄灯末期的时间也不能得到充分利用，造成一定的时间损失，记为 $loss2$ 。这样某一个相位的有效绿灯时间就等于绿灯时间和黄灯时间的总和再去掉可通行时间前后的损失时间：

$$T_g = T_{green} + T_{yellow} - loss1 - loss2 \tag{4.7}$$

实际情况下黄灯时间一般取 2-5 秒，我们在此假定黄灯时间为 3s。设交叉口长度约 20m，车流的平均速度为 10m/s，从停止加速到平均速度的时间为 3s，加速损失时间计算则为 1.5s，信号灯转换时间为 0.5s，驾驶员反应时间和松手刹、挂挡时间为 1s，黄灯之后为了保证安全而损失的可通行时间约为 1.5s，再加上意外情况可能导致的损失时间，则每个相位的损失时间

$$loss1 + loss2 \approx 5s \tag{4.8}$$

考虑四个相位的情况，则总周期 T 可表示为：

$$T = \sum_2^4 (T_{green(i)} + T_{yellow(i)}) \quad (4.9)$$

若用有效绿灯时间 T_g 来表示, 则为:

$$T = T_{g1} + T_{g2} + T_{g3} + T_{g4} + L \quad (4.10)$$

其中 $L = \sum (loss1_i + loss2_i)$ 为一个周期总的损失时间。

在车辆到达率为常数的情况下, 车辆的阻滞延误与车辆到达率的关系是一种线性关系。如图 4.4 所示, 车辆 A 到达停车线或到达等候车队的队尾时, 正值红灯期间, 此时在它的前面已有先期到达的 N 辆车在停车线后等候。因此这辆车 A 要等到绿灯亮之后, 且它前面的 N 辆车全部离开停车线之后它才能驶出停车线, 而不是绿灯亮后马上可以通过交叉口。

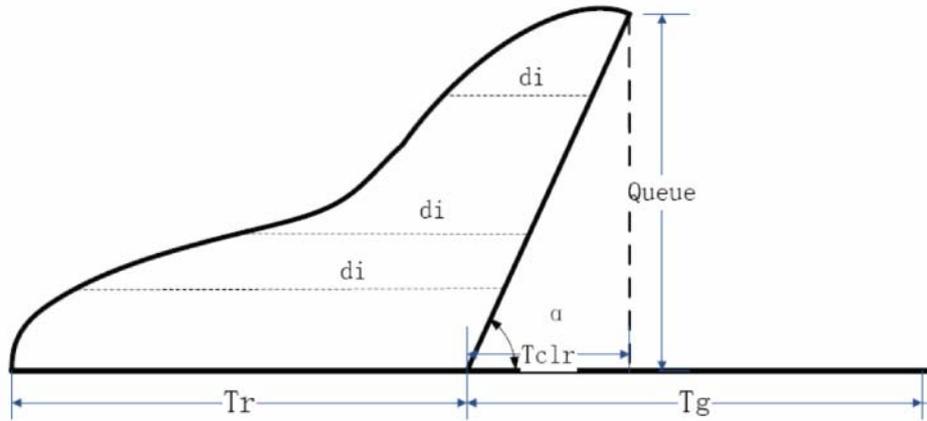


图 4.4 总延误时间的说明

从图 4.4 可以看出, 所有车辆的延误时间曲线近似与一个三角形。三角形的每一条水平线为每一辆车的延误时间, 而三角形的垂直线代表不同瞬时车辆的排队长度。所以在一个信号周期内, 该相位的全部车辆的总延误时间等于三角形的面积, 即每辆车延误时间的总和, 该面积为:

$$S_{\Delta} = \frac{1}{2} T_r \cdot Queue \quad (4.11)$$

其中 T_r 为红灯的等待时间, $Queue$ 为排队的车辆刚被清空时一共通过的车辆数,

从途中又可看出 $Queue = T_{clr} \cdot \tan(\alpha)$, T_{clr} 为清空队列的时间, α 角度所表示的斜率代表的是队列前部被清空的速率, 也就是饱和流量 s , 所以上图的面积又可写成:

$$S_{\Delta} = \frac{1}{2} T_r \cdot T_{clr} \cdot s \quad (4.12)$$

T_{clr} 队列的清空时间取决于绿灯刚亮时队列中已经存在的车辆数, 可以用 qT 来表示, 而且队列前部不断被清空时, 队列后部也有新到达的车辆补充进来, 速率为流量 q , 所以清空时间的公式为:

$$T_{clr} = \frac{q \cdot T_r}{s - q} \quad (4.13)$$

代入面积公式得：

$$S_{\Delta} = \frac{T_r^2 \cdot q \cdot s}{2(s - q)} = \frac{T_r^2 q}{2(1 - q/s)} = \frac{(T - T_g^2)q}{2(1 - q/s)} \quad (4.14)$$

此处 $S_{\Delta} = \frac{T_r^2 \cdot q \cdot s}{2(s - q)}$ 与上节公式 4.6 相比较可见，4.6 中的 t 其实就是红灯时间 T_r ，于

是可见 λ 其实就对应了修正后的流量 $\frac{q \cdot s}{(s - q)}$ ，这就是修正后考虑了排队问题的流量值。

式 4.14 中的 T_g 为有效绿灯时间，计算出来的面积表征该相位一个周期内所有车辆总的延误时间。该相位一个周期总的车辆数为 $q \cdot T$ ，因此平均每辆车的延误时间为：

$$d = \frac{(T - T_g^2)}{2T(1 - q/s)} \quad (4.15)$$

以上的计算是基于车辆均匀到达交叉口这一前提而引起的延误时间，一般称为均匀延误时间。经统计数据发现，这样计算出来的延误时间偏低，因为它没有考虑车辆到达的随机性。可以这样解释这种现象：除红灯期间车辆排队引起的延误外，还有一种随机延误，这种随机延误可以用排队论来描述，把交通信号作为服务台，到达的车辆作为顾客。根据韦伯斯特的研究^[3]，随机波动所产生的附加延误近似可表达为 $\frac{x^2}{2q(1-x)}$ ，这里

的 x 表示饱和度，定义为 $x = \frac{qT}{sT_g}$ 它的物理意义是当饱和度过大时，在一个绿灯时段内

可能还不能够清空排队的序列。因为之前计算得 $T_{clr} = \frac{q \cdot T_r}{s - q}$ ，若要使 $T_{clr} < T_g$ ，则有：

$$\begin{aligned} T_g > \frac{q \cdot T_r}{s - q} &\Rightarrow \frac{T_r}{T_g} < \frac{s - q}{q} \\ &\Rightarrow \frac{T_r + T_g}{T_g} < \frac{s - q + q}{q} \Rightarrow \frac{T}{T_g} < \frac{s}{q} \Rightarrow \frac{qT}{sT_g} < 1 \end{aligned} \quad (4.16)$$

这里 $\frac{qT}{sT_g}$ 就是定义的饱和度 x ，即饱和度小于 1 才能保证排队的车辆在一个周期内能够及时地通过路口。

将平均延误时间和随机延误的附加时间加在一起，就得到该相位一周期内每辆车的

平均延误时间:

$$d = \frac{(T - T_g)^2}{2T(1 - q/s)} + \frac{x^2}{2q(1 - x)} \quad (4.17)$$

则对于十字路口的四个相位总延误时间为

$$D = \sum_1^4 d_i q_i = \sum_1^4 \frac{T q_i (1 - T_{gi}/T)^2}{2(1 - q_i/s_i)} + \frac{x_i^2}{2(1 - x_i)}$$

此即为优化的目标函数, 参看基本假设, 其要满足的约束条件为:

$$\begin{cases} T_{gi} \geq 10 \\ T \leq 200 \\ \sum_1^4 T_{gi} + L = T \\ 0.7 \leq x \leq 0.9 \end{cases} \quad (4.18)$$

对于丁字路口我们可得:

$$D = \sum_1^3 d_i q_i = \sum_1^3 \frac{T q_i (1 - T_{gi}/T)^2}{2(1 - q_i/s_i)} + \frac{x_i^2}{2(1 - x_i)} \quad (4.19)$$

$$s.t. \begin{cases} T_{gi} \geq 10 \\ T \leq 200 \\ \sum_1^4 T_{gi} + L = T \\ 0.7 \leq x \leq 0.9 \end{cases} \quad (4.20)$$

4.2.3 结果分析

为了检验上述算法的优越性, 我们采用韦伯斯特经验公式^[3]:

$$T = \frac{1.5L + 5}{1 - \sum q_i/s_i} \quad (4.21)$$

对同一个路口进行规划, 并把所得的等待时间与我们的算法进行比较, 现假设主干道流量为 1300 辆/小时, 饱和流量为 8000 辆/小时, 支路流量为 500 辆/小时, 饱和流量为 4000 辆/小时。即对应的程序输入向量为 $P = [1300; 500; 8000; 4000]$, 分别采用两个算法对十字型路口和丁字型路口进行计算, 计算结果如下:

表 4.1 十字路口点控制

	第一相位绿 信比	第二相位绿 信比	第三相位绿 信比	第四相位绿 信比	周期 (s)	单位时间内 所有车平均 等待时间 (s)
上述算法	0.2033	0.1990	0.1562	0.1551	69.8400	26.9959
韦伯斯特	0.2140	0.2140	0.1646	0.1646	82.3529	28.0909

表 4.2 丁字路口点控制

	第一相位绿信比	第二相位绿信比	第三相位绿信比	第四相位绿信比	周期 (s)	单位时间内所有车平均等待时间 (s)
上述算法	0.2469	0.2469	0.1931	0.0000	47.8998	12.8615
韦伯斯特	0.2528	0.2528	0.1944	0.0000	50.0000	13.0543

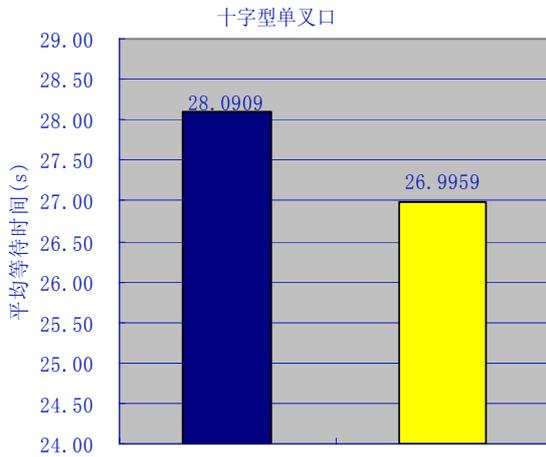


图 4.5 十字型路口平均等待时间

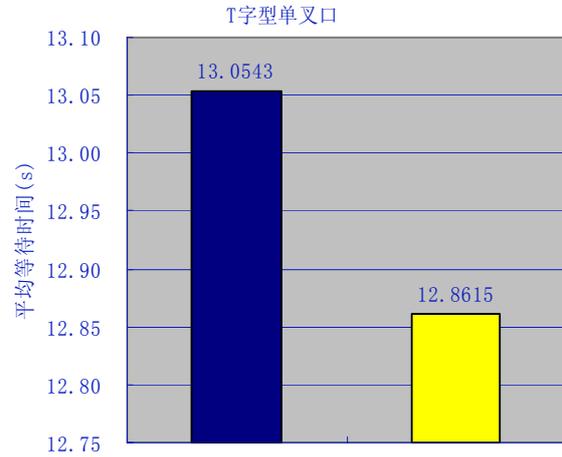


图 4.6 丁字型路口平均等待时间

由图可知，我们的新算法相对于韦伯斯特算法有明显的优势。

4.3 多交叉口（线）实时控制模型

4.3.1 基本参数

由于两个相邻路口之间联系的复杂性，我们用“绿波带理论”来进行简化分析，详细内容如下：

1) 多个交叉路口的布局

考虑分布三个交叉口 A、B、C 的道路，交叉口间距分别为 11、12，如图 4.5 所示。

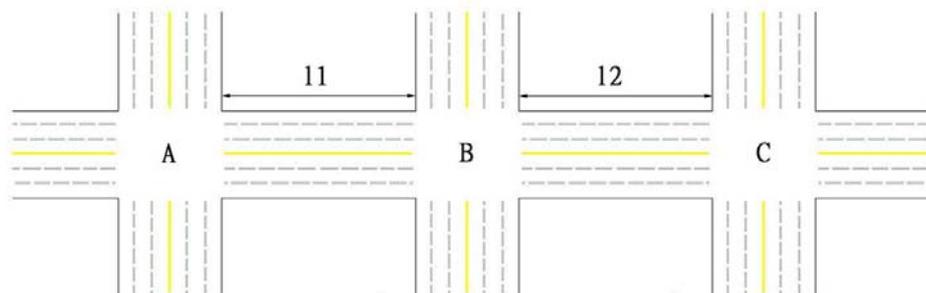


图 4.7 线状 3 交叉路口的分布

2) 周期长、绿信比和相位差的选取^[1]

在线控系统中，为了使各交叉口的交通信号能取得协调，各叉路口的交通信号周期必须统一。为此必须先按单交叉路口信号控制的配时方法，计算其周期长度，然后从中选取最长周期时作为这个系统的公共周期。

线控系统中各叉路口信号的绿信比不一定相同，通常要根据每个交叉路口各方向的交通流量比确定。

相位差直接决定干线控制系统的有效性，显然相位差与相邻叉路口间的距离以及车

辆在其间的行驶速度有关。

3) 图解法^[4]

按照交通控制论,线控制系统配时方案通常可用时间-距离图来描述,如图 4.6 所示。图中以时间(即信号配时)为纵坐标,干道上交叉口间距为横坐标。

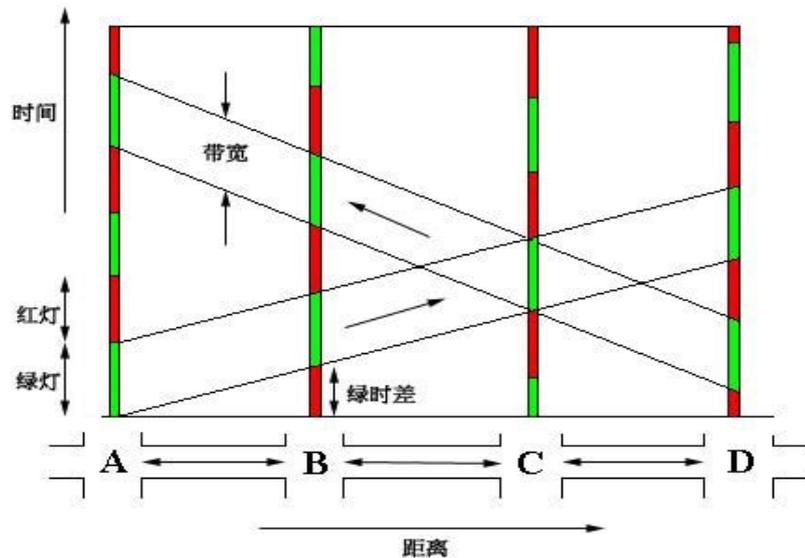


图 4.8 时间-距离图

在时间-距离图上协调线控制系统的时差,同时调整确定通过带速度和周期时长,如图 4.7 所示。相邻五个交叉口(A, B, C, D, E),纳入一个线控系统,根据调整系统通过带速度在 36km/h ,按上述方法,相应的系统周期时长暂定为 60s 。图中各竖线上的粗线段表示红灯时段,如 A 交叉口竖线 AA'上的 1—2、3—4、5—8 段;细线表示绿灯时段。

(1) 从 4 点引一相当于 36km/h 带速的斜线①,此斜线与 BB'线的交点,同从 AA'上 1 点所引水平线同 BB'线的交点(BB'线上的 1 点)很接近。BB'上的 1 点可取为 B 交叉口同 4 交叉口配成交互式协调的绿时差;在 BB'线上相应于 AA'线画出 2—3、4—5 粗线段,为交叉口的红灯时段。

(2) 连接 4 点和 BB'上的 1 点成斜线②,线②同 CC'的交点,同从 AA'上 2 点所引水平线同 CC'的交点(CC'上的 2 点)很接近,CC'上的 2 点也可以取为 C 交叉口对 B 交叉口组成交互式协调的绿时差,所以在 CC'竖线上可画 1—2、3—4、5—6 各粗线段,为 C 交叉口的红灯时段。

(3) 连接 4 点和 CC'的 2 点成斜线③,线③在 DD'上的交点同从 AA'上 2 点所引水平线同 DD'的交点(DD'上的 2 点)很接近,所以 C 交叉口与 D 交叉口应是同步式协调,在 DD'上画与 CC'相同的 1—2、3—4、5—6 红灯时段。

(4) 以下用同样的方法在 EE'线上作出红灯时段。这样就配成各交叉口由交互式与同步式组合成的双向线控制系统。

(5) 在图上作最后的通过带,算得带速约 57km/h ,带宽 16s ,为周期时长 60s 的 27% 。这样的带速和实际车速相比过高,为了降低带速,有必要相应加长周期时长,为使带速控制在 40km/h 左右,延长周期时长为 $85\text{—}90\text{s}$ 。

(6) 调整绿信比,实际上各交叉口的绿信比都不相同,可用以下简单方法调整:不移动按上述方法求得的各交叉口的红灯(或绿灯)的中心位置,只将红灯(或绿灯)的时间按实际绿信比延长或缩短即可。经这样调整后,通过带宽增加不少,但仍低于 50% 周期时长。

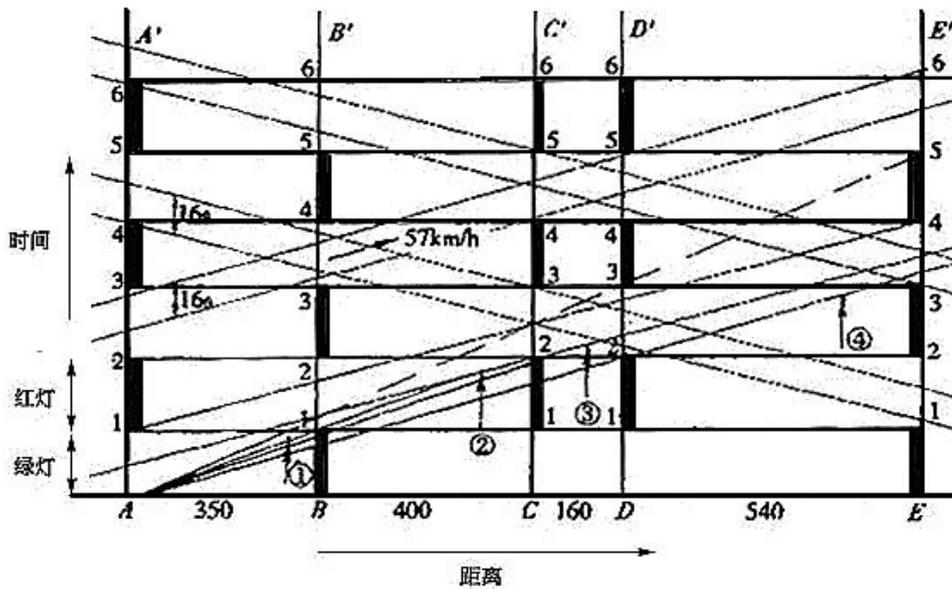


图 4.9 协调时差图解法示例

4.3.2 模型建立

以 $A \rightarrow B \rightarrow C$ 作为主干道，车流最大，确保车辆主干道上行驶时，能根据绿波带理论通常无阻，其他因为右转，左转而进入主干道的车辆绿灯同样畅行，红灯后，它们以及主干道上的车辆的排空时间应该被考虑。

1) 传统优化方法

首先，把三个叉路口分开考虑，分别使用点控制算法对它们进行优化，最繁忙的叉路口的周期也会最大，以此最大周期为线控制的统一周期；其次，按照上述“绿波带”控制方法对各个叉路口的绿信比进行优化，得出相应的延误时间。

可得数学模型为：

$$\text{目标函数} \quad D = \sum_{j=1}^3 \sum_{i=1}^4 d_{ij} q_{ij} = \sum_{j=1}^3 \sum_{i=1}^4 \frac{T q_{ij} (1 - T_{gij} / T_j)^2}{2(1 - q_{ij} / s_{ij})} + \frac{x_{ij}^2}{2(1 - x_{ij})} \quad (4.22)$$

$$\text{s.t.} \quad \begin{cases} T_{gij} \geq 10 \\ T_j = T \\ \sum_1^4 T_{gij} + L_j = T_j \\ 0.7 \leq x_{ij} \leq 0.9 \end{cases} \quad (4.23)$$

$$\text{由上式可得系统周期} \quad T = \max(T_j) \quad (4.24)$$

$$\text{相位计算} \quad \text{phase}_j = \frac{\omega}{v} + \frac{q_{11}(T - T_{g11})}{s_{11} - q_{11}} - \frac{q_{1j}(T - T_{g1j})}{s_{1j} - q_{1j}}$$

2) 改进优化方式

改进优化算法在传统优化方法的基础上采用二次优化，即在按点控制算法得出三叉

口的最大周期，并按绿波带要求进行常规优化后，我们再以此对每个交叉口用优化算法分别进行优化，更好保证优化的实时性。二次优化模型为

$$D = \sum_{j=1}^3 \sum_{i=1}^4 d_{ij} q_{ij} = \sum_{j=1}^3 \sum_{i=1}^4 \frac{T q_{ij} (1 - T_{gij} / T)^2}{2(1 - q_{ij} / s_{ij})} + \frac{x_{ij}^2}{2(1 - x_{ij})} \quad (4.25)$$

$$s.t. \quad \left\{ \begin{array}{l} T_{gij} \geq 10 \\ T_j \leq 200 \\ \sum_1^4 T_{gij} + L_j = T_j \\ 0.7 \leq x_{ij} \leq 0.9 \\ T_j = T \end{array} \right. \quad (4.26)$$

4.3.3 结果分析

分别用上述两种方法进行计算，程序初始输入向量仍为 $P = [1300; 500; 8000; 4000]$ ，计算结果为：

表 4.3 线控制结果

	第一相位 绿信比	第二相位 绿信比	第三相位 绿信比	第四相位 绿信比	周期 (s)	单位时间 内所有车 平均等待 时间 (s)
传统方法	0.2242	0.2207	0.1512	0.1505	78.9129	90.2898
	0.3119	0.1978	0.1185	0.1185	78.9129	
	0.2949	0.1761	0.1383	0.1373	78.9129	
改进方法	0.2242	0.2207	0.1512	0.1505	78.9129	81.7527
	0.2334	0.2334	0.1398	0.1398	78.9129	
	0.2167	0.2070	0.1625	0.1604	78.9129	

表 4.4 线控制相位差

	第 1 叉口	第 2 叉口	第 3 叉口
相位差 (s)	0.0000	33.4522	52.1331

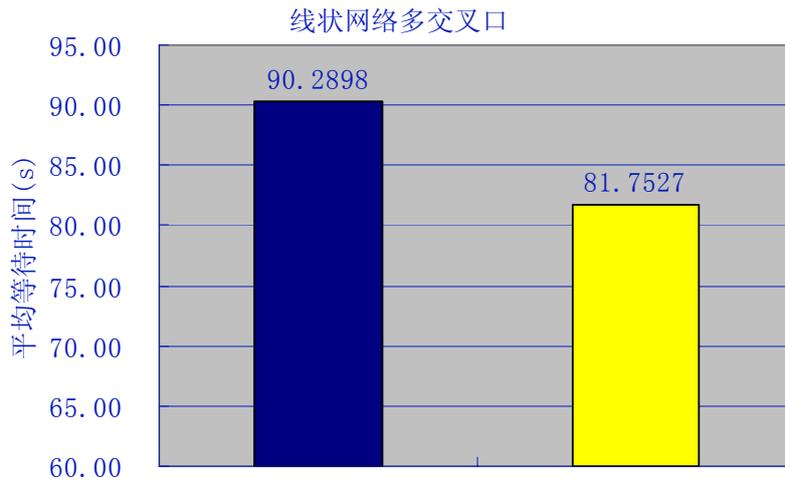


图 4.10 线状网络平均等待时间

通过比较传统优化与改进的优化配时方法，可以明显看出，改进的优化方法通过实时优化各叉口绿信比，可以更加有效的减小车辆延误时间。

4.4 多交叉口（面）实时控制模型

4.4.1 基本参数

1) 多交叉路口的选取

考虑分布六个交叉口 A、B、C、D、E、F 的平面区域，交叉口间距分别为 11、12、13、14、15、16，如图 4.6 所示。

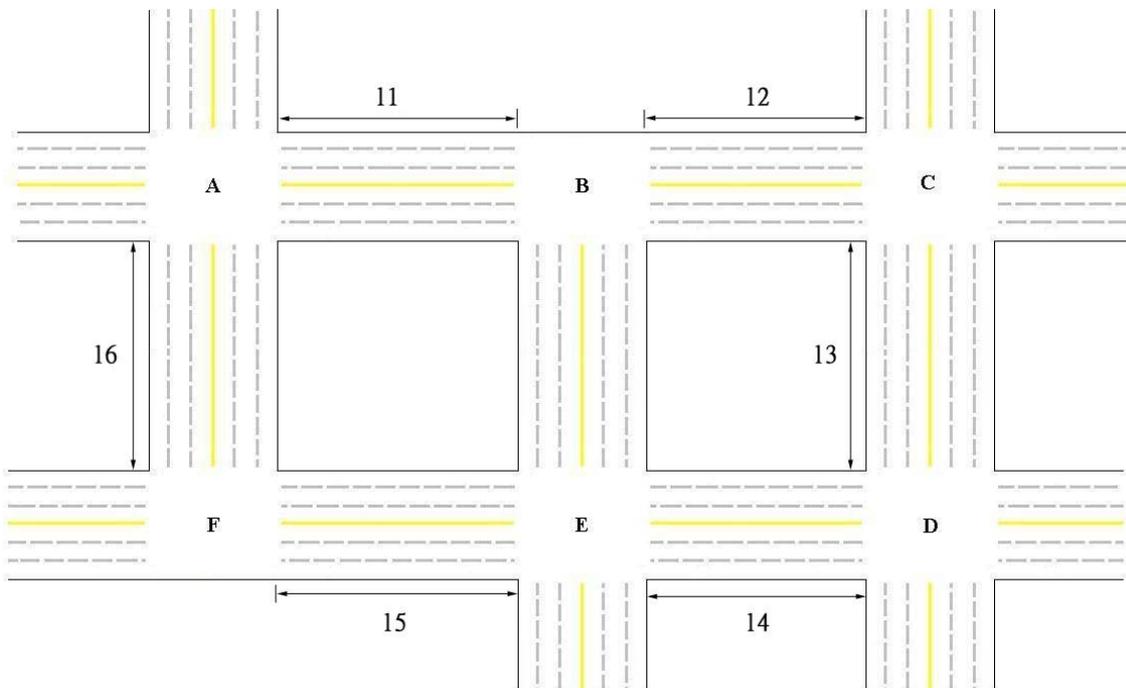


图 4.11 交叉路口的网络分布

4.4.2 模型建立

以 $A \rightarrow B \rightarrow C; C \rightarrow D; D \rightarrow E \rightarrow F$ 为主干道，使用绿波带理论优化，另两条支路以及节点

路口在此基础上进行二重优化。

首先对六个叉口分别进行点优化，取出最繁忙叉路口（A 叉口）的最大周期作为整个网络的共同周期，对主干道 $A \rightarrow B \rightarrow C$ 进行线控制，以此确定 B, C 的结果，再在此基础上对 $C \rightarrow D$ 进行线优化，继而在 D 的基础上对主干道 $D \rightarrow E \rightarrow F$ 进行优化，最后对总体进行优化，可得数学模型如下

$$\text{目标函数} \quad D = \sum_{j=1}^6 \sum_{i=1}^4 d_{ij} q_{ij} = \sum_{j=1}^6 \sum_{i=1}^4 \frac{T q_{ij} (1 - T_{gij} / T_j)^2}{2(1 - q_{ij} / s_{ij})} + \frac{x_{ij}^2}{2(1 - x_{ij})} \quad (4.27)$$

$$\text{s.t.} \quad \begin{cases} T_{gij} \geq 10 \\ T_j = T \\ \sum_1^4 T_{gij} + L_j = T_j \\ 0.7 \leq x_{ij} \leq 0.9 \end{cases} \quad (4.28)$$

$$\text{由上式可得系统周期} \quad T = \max(T_j) \quad (4.29)$$

$$\text{相位计算} \quad \text{phase}_j = \frac{\omega}{v} + \frac{q_{11}(T - T_{g11})}{s_{11} - q_{11}} - \frac{q_{1j}(T - T_{g1j})}{s_{1j} - q_{1j}} \quad (4.30)$$

$$\text{总体优化模型为} \quad D = \sum_{j=1}^6 \sum_{i=1}^4 d_{ij} q_{ij} = \sum_{j=1}^6 \sum_{i=1}^4 \frac{T q_{ij} (1 - T_{gij} / T)^2}{2(1 - q_{ij} / s_{ij})} + \frac{x_{ij}^2}{2(1 - x_{ij})} \quad (4.31)$$

$$\text{s.t.} \quad \begin{cases} T_{gij} \geq 10 \\ T_j \leq 200 \\ \sum_1^4 T_{gij} + L_j = T_j \\ 0.7 \leq x_{ij} \leq 0.9 \\ T_j = T \end{cases} \quad (4.32)$$

4.4.3 结果分析

我们在此处仍然分别采用了传统优化方法和改进优化方法，并对两种方法获得的结果进行了比较

表 4.5 面控制结果

	第一相位 绿信比	第二相位 绿信比	第三相位 绿信比	第四相位 绿信比	周期 (s)	单位时间 内所有车 平均等待 时间 (s)
传统方法	0.2242	0.2207	0.1512	0.1505	78.9129	163.2866
	0.4944	0.2023	0.1132	0.0000	78.9129	
	0.2618	0.1971	0.1442	0.1434	78.9129	
	0.3119	0.1978	0.1185	0.1185	78.9129	
	0.2949	0.1761	0.1383	0.1373	78.9129	

	0.5415	0.1757	0.0928	0.0000	78.9129	
改进方法	0.2242	0.2207	0.1512	0.1505	78.9129	145.2690
	0.3172	0.3172	0.1754	0.0000	78.9129	
	0.2202	0.2141	0.1568	0.1555	78.9129	
	0.2334	0.2334	0.1398	0.1398	78.9129	
	0.2167	0.2070	0.1625	0.1604	78.9129	
	0.3520	0.3520	0.1058	0.0000	78.9129	

表 4.6 面控制相位差

	第 1 叉口	第 2 叉口	第 3 叉口	第 4 叉口	第 5 叉口	第 6 叉口
相位差 (s)	0.00	31.69	51.07	68.45	87.13	119.20

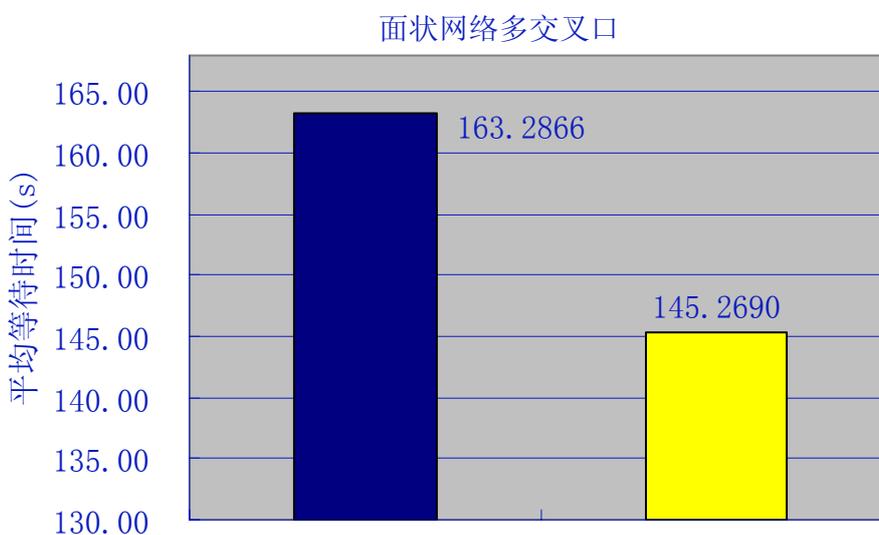


图 4.12 面状网络平均等待时间

由图亦不难发现，对于面控制，改进算法同样明显优于传统算法。

五、基于泊松分布的实时交通流序列

5.1 实时交通序列的产生

A、车辆到达分布

我们用概率论中描述可数事件统计特性的离散型分布做为工具考虑车辆到达的随机性分布,考察在一段固定长度的时间内到达的交通量波动。:考虑到如果车流密度不大,车辆间的相互影响可以忽略,车辆基本上为自由行驶,结合本文中假设的道路条件,采用泊松(Poisson)分布:

$$P(x) = \frac{(\lambda t)^x}{x!} e^{-\lambda t}, \quad x = 0, 1, 2, \dots \quad (5.1)$$

其中:

$P(x)$ 为在计数周期 t 内到达 x 辆车的概率; λ 为单位时间的车辆平均到达率; t 为每个计数周期的间隔时间; e 为自然对数的底, 取值 2.718280

若令 $m = \lambda t$ ， m 为在计数周期 t 内平均到达的车辆数，则式(3-1)可写为

$$P(x) = \frac{(m)^x}{x!} e^{-m}, \quad x = 0, 1, 2, \dots \quad (5.2)$$

B、实时序列

在初始时刻 T_0 ，按实际测量值给定一个流量值，有这个流量值后，实时算法会产生周期和绿信比，从而可以得到这个周期内的红灯时间 T_r ，即等待时间。等待时间确定后，为了给控制算法一个实时的交通流环境，必须给出在下一个周期开始时已经到达路口的车辆数量，由上述讨论知该值满足离散的泊松分布，且泊松分布的 m 受流量和等待时间影响。假设车流量在此路口基本不会突变，我们使用本周期 T 内的流量和等待时间 T_r 来产生下一周期 T_1 泊松分布的 m 值，再按此 m 产生泊松分布，最后由此泊松分布随机依次产生下个周期开始时的车量，该随机量能模拟当时路口的情况，并以此类推，实时产生交通流序列。整个流程的数学模型如下，

设上文的优化算法函数为：

$$(T_{(k+1)gij}, T_{k+1}) = f(p_k) \quad (5.3)$$

其中 $T_{(k+1)gij}$ 为地 $k+1$ 个周期有效绿灯时间， T_{k+1} 为第 $k+1$ 个周期， p_k 为第 k 个周期各路口流量和饱和流量向量，如上文中的 $P = [1300; 500; 8000; 4000]$ 表示主干道流量为 1300 辆/小时，饱和流量为 8000 辆/小时，支路流量为 500 辆/小时，饱和流量为 4000 辆/小时。

$$\text{则可认为:} \quad T_{(k+1)r} = T_{k+1} - T_{(k+1)gij} \quad (5.4)$$

$$\text{所以可得下一周期泊松分布的 } m \text{ 为} \quad m_{k+2} = T_{(k+1)r} \times p_k \quad (5.5)$$

由此可随机获得第 $k+2$ 个周期时的车量 Q_{k+2} ，继而由 $Q_{k+2} / T_{(k+1)r}$ 获得 p_{k+1} ，再由

$(T_{(k+2)gij}, T_{k+2}) = f(p_{k+1})$ 计算下一周期的 T_r 和 T 。按此方法可获得如下车量实时序列：

表 5.1 随机产生的车量

	第 1 周期	第 2 周期	第 3 周期	第 4 周期	第 5 周期
第 1 相位	13	16	15	11	17
第 2 相位	10	13	10	9	15
	第 6 周期	第 7 周期	第 8 周期	第 9 周期	第 10 周期
第 1 相位	15	11	17	16	13
第 2 相位	14	11	13	14	16

在此产生的数据环境中使用点优化算法可得如下结果：

表 5.2 随机车辆序列下的点控制

	第 1 周期	第 2 周期	第 3 周期	第 4 周期	第 5 周期
--	--------	--------	--------	--------	--------

第 1 相位	0.1786	0.1426	0.1432	0.1301	0.1325
第 2 相位	0.1786	0.1426	0.1432	0.1301	0.1325
第 3 相位	0.1786	0.2193	0.2556	0.1996	0.2357
第 4 相位	0.1786	0.2193	0.2556	0.1996	0.2357
周期(s)	70.0000	72.3921	98.8788	58.7164	75.8447
	第 6 周期	第 7 周期	第 8 周期	第 9 周期	第 10 周期
第 1 相位	0.1378	0.1051	0.1153	0.1322	0.1294
第 2 相位	0.1378	0.1051	0.1153	0.1322	0.1294
第 3 相位	0.2760	0.2335	0.2692	0.2448	0.2603
第 4 相位	0.2760	0.2335	0.2692	0.2448	0.2603
周期(s)	116.0031	61.9494	86.5397	81.3614	90.7007

5.2 不同配时方案的比较

在上文产生的随机泊松车量实时序列下，运用定周期定绿信比，定周期变绿信比，以及变周期变绿信比进行规划，结果如下：

表 5.3 定周期定绿信比

	第 1 周期	第 2 周期	第 3 周期	第 4 周期	第 5 周期
第 1 相位	0.2033	0.2033	0.2033	0.2033	0.2033
第 2 相位	0.1990	0.1990	0.1990	0.1990	0.1990
第 3 相位	0.1562	0.1562	0.1562	0.1562	0.1562
第 4 相位	0.1551	0.1551	0.1551	0.1551	0.1551
周期(s)	69.8400	69.8400	69.8400	69.8400	69.8400
	第 6 周期	第 7 周期	第 8 周期	第 9 周期	第 10 周期
第 1 相位	0.2033	0.2033	0.2033	0.2033	0.2033
第 2 相位	0.1990	0.1990	0.1990	0.1990	0.1990
第 3 相位	0.1562	0.1562	0.1562	0.1562	0.1562
第 4 相位	0.1551	0.1551	0.1551	0.1551	0.1551
周期(s)	69.8400	69.8400	69.8400	69.8400	69.8400

表 5.4 定周期变绿信比

	第 1 周期	第 2 周期	第 3 周期	第 4 周期	第 5 周期
第 1 相位	0.2033	0.1722	0.1569	0.1554	0.1308
第 2 相位	0.1990	0.1694	0.1555	0.1528	0.1308
第 3 相位	0.1562	0.1867	0.2011	0.2037	0.2260
第 4 相位	0.1551	0.1853	0.2002	0.2018	0.2260
周期(s)	69.8400	69.8400	69.8400	69.8400	69.8400
	第 6 周期	第 7 周期	第 8 周期	第 9 周期	第 10 周期
第 1 相位	0.1303	0.1294	0.1265	0.1246	0.1289
第 2 相位	0.1303	0.1294	0.1265	0.1246	0.1289
第 3 相位	0.2265	0.2274	0.2303	0.2323	0.2279
第 4 相位	0.2265	0.2274	0.2303	0.2323	0.2279
周期(s)	69.8400	69.8400	69.8400	69.8400	69.8400

表 5.5 变周期变绿信比

	第 1 周期	第 2 周期	第 3 周期	第 4 周期	第 5 周期
第 1 相位	0.2033	0.1737	0.1626	0.1558	0.1214
第 2 相位	0.1990	0.1705	0.1602	0.1530	0.1214
第 3 相位	0.1562	0.1879	0.2068	0.2040	0.2553
第 4 相位	0.1551	0.1864	0.2052	0.2020	0.2553
周期(s)	69.8400	71.0637	75.4126	70.1318	81.0796
	第 6 周期	第 7 周期	第 8 周期	第 9 周期	第 10 周期
第 1 相位	0.1299	0.1177	0.1237	0.1114	0.1161
第 2 相位	0.1299	0.1177	0.1237	0.1114	0.1161
第 3 相位	0.2261	0.2638	0.2286	0.2739	0.2714
第 4 相位	0.2261	0.2638	0.2286	0.2739	0.2642
周期(s)	69.4734	84.4061	67.7108	87.2377	86.1286

总等候时间为：

表 5.6 总等候时间

定周期定绿信比	定周期变绿信比	变周期变绿信比
388.3571	230.8897	227.4470

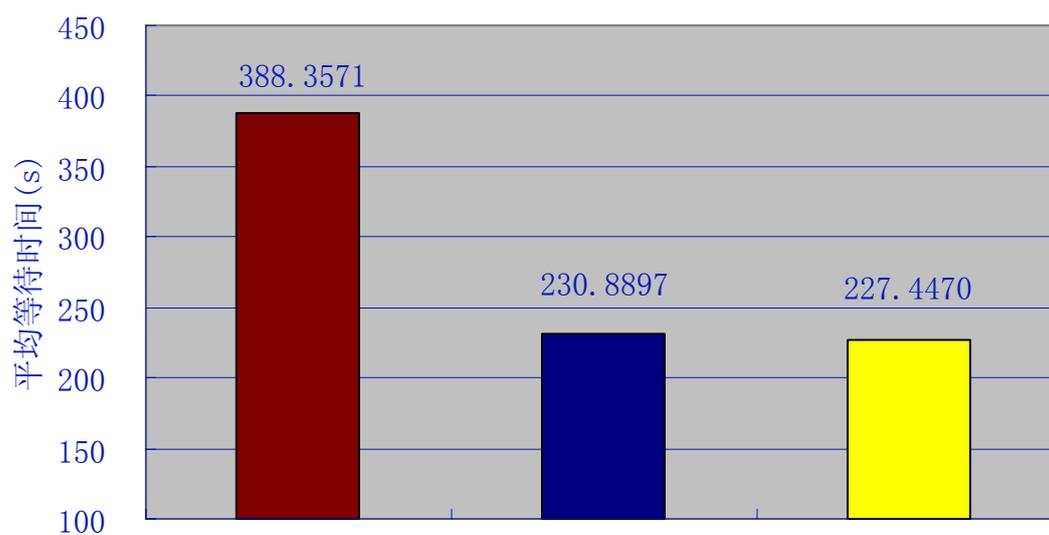


图 5.1 各种控制方法的比较

明显可见变周期变绿信比的方法是最优的，其次是变周期定绿信比，最不好的是定周期定绿信比的情况。

5.2 线控制的实时配时方案

在随机产生的车流量条件下，对线状区域的各个交叉口进行优化计算，可得周期与绿信比如下表所示。

表 5.7 线状区域周期及绿信比

	第 1 周期			第 2 周期		
	第 1 叉口	第 2 叉口	第 3 叉口	第 1 叉口	第 2 叉口	第 3 叉口
第 1 相位	0.224231	0.233436	0.216654	0.178731	0.155384	0.16627
第 2 相位	0.220677	0.233436	0.207001	0.178731	0.152334	0.16627
第 3 相位	0.151186	0.139843	0.162474	0.172965	0.198876	0.185426
第 4 相位	0.150462	0.139843	0.160427	0.172965	0.196798	0.185426
周期(s)	78.91293	78.91293	78.91293	67.42903	67.42903	67.42903
相位差(s)	0	33.45225	52.13311	0	30.21391	51.01722

	第 3 周期			第 4 周期		
	第 1 叉口	第 2 叉口	第 3 叉口	第 1 叉口	第 2 叉口	第 3 叉口
第 1 相位	0.157537	0.11783	0.171527	0.11737	0.11448	0.169451
第 2 相位	0.155807	0.11783	0.171527	0.116295	0.11448	0.169451
第 3 相位	0.226227	0.266728	0.210673	0.270223	0.271484	0.216513
第 4 相位	0.224768	0.261951	0.210611	0.26804	0.271484	0.216513
周期(s)	84.86784	84.86784	84.86784	87.69152	87.69152	87.69152
相位差(s)	0	33.5586	51.09111	0	31.48699	50.90066

	第 5 周期		
	第 1 叉口	第 2 叉口	第 3 叉口
第 1 相位	0.086512	0.115901	0.126224
第 2 相位	0.086512	0.115901	0.126224
第 3 相位	0.313774	0.293701	0.274061
第 4 相位	0.313774	0.275067	0.274061
周期(s)	100.286	100.286	100.286
相位差(s)	0	27.2299	47.40519

5.3 面控制的实时配时方案

在随机产生的车流量条件下，对网状区域的各个交叉口进行优化计算，可得周期与绿信比如下表所示。

表 5.8 面状网络周期及绿信比

	第 1 周期					
	第 1 叉口	第 2 叉口	第 3 叉口	第 4 叉口	第 5 叉口	第 6 叉口
第 1 相位	0.224231	0.317237	0.220227	0.233436	0.216654	0.35204
第 2 相位	0.220677	0.317237	0.214088	0.233436	0.207001	0.35204
第 3 相位	0.151186	0.175443	0.15676	0.139843	0.162474	0.105837
第 4 相位	0.150462	0	0.15548	0.139843	0.160427	0
周期(s)	78.91293	78.91293	78.91293	78.91293	78.91293	78.91293
相位差(s)	0	31.69371	51.07459	68.45225	87.13311	119.2061

第 2 周期						
	第 1 叉口	第 2 叉口	第 3 叉口	第 4 叉口	第 5 叉口	第 6 叉口
第 1 相位	0.16632	0.278191	0.156515	0.168606	0.155483	0.258846
第 2 相位	0.161605	0.278191	0.154115	0.16365	0.153178	0.258846
第 3 相位	0.204382	0.243976	0.212504	0.202214	0.213472	0.282666
第 4 相位	0.201504	0	0.210677	0.199341	0.211678	0
周期(s)	75.1346	75.1346	75.1346	75.1346	75.1346	75.1346
相位差(s)	0	29.97076	49.94786	64.9019	84.98374	114.1482

第 3 周期						
	第 1 叉口	第 2 叉口	第 3 叉口	第 4 叉口	第 5 叉口	第 6 叉口
第 1 相位	0.10263	0.256252	0.149392	0.147339	0.181662	0.266979
第 2 相位	0.10263	0.252699	0.142965	0.147339	0.181662	0.266979
第 3 相位	0.29474	0.337104	0.254987	0.25003	0.215708	0.312096
第 4 相位	0.29474	0	0.247396	0.25003	0.215708	0
周期(s)	97.43722	97.43722	97.43722	97.43722	97.43722	97.43722
相位差(s)	0	28.00555	46.88794	63.18331	81.56195	110.3933

第 4 周期						
	第 1 叉口	第 2 叉口	第 3 叉口	第 4 叉口	第 5 叉口	第 6 叉口
第 1 相位	0.098465	0.208661	0.109853	0.123804	0.139122	0.24245
第 2 相位	0.098465	0.208661	0.109853	0.123804	0.139117	0.242711
第 3 相位	0.291643	0.41784	0.280255	0.2737	0.250989	0.350001
第 4 相位	0.291643	0	0.280255	0.25891	0.250989	0
周期(s)	90.99877	90.99877	90.99877	90.99877	90.99877	90.99877
相位差(s)	0	27.45635	50.255	62.8346	84.70936	112.1308

第 5 周期						
	第 1 叉口	第 2 叉口	第 3 叉口	第 4 叉口	第 5 叉口	第 6 叉口
第 1 相位	0.154884	0.30107	0.10365	0.122198	0.123027	0.278705
第 2 相位	0.154884	0.171887	0.10365	0.122198	0.123027	0.275318
第 3 相位	0.222089	0.342502	0.273322	0.254774	0.255597	0.261435
第 4 相位	0.222089	0	0.273322	0.254774	0.252293	0
周期(s)	81.28266	81.28266	81.28266	81.28266	81.28266	81.28266
相位差(s)	0	32.04105	52.32224	67.79655	84.84975	117.0814

六、算法复杂性、可计算性分析及改进

对于单个叉路口的点控制，尚属简单的，确定完目标函数和约束条件后进行优化求解就可以了，但对于线控制和面控制建模存在很大困难，因为在系统中各个叉口之间的联系很难确定，为了简化计算，我们使用了“绿波带理论”，并在此基础上进行了优化，同时，使用泊松分布随机产生车流实时序列，虽然所有算法都可计算，我们也成功的编写了所有程序，但由于运算量较大，在运算时会出现电脑“卡住”很多时间，且对于这

个算法最优性明确而严密的证明尚有待进一步讨论。

受系统控制输入-输出模型的启发，我们试图引入另一种算法，由于时间有限，这种算法我们无法深入研究，在此仅做介绍，我们以三个交叉路口的线控制为例，如图 5.2 所示

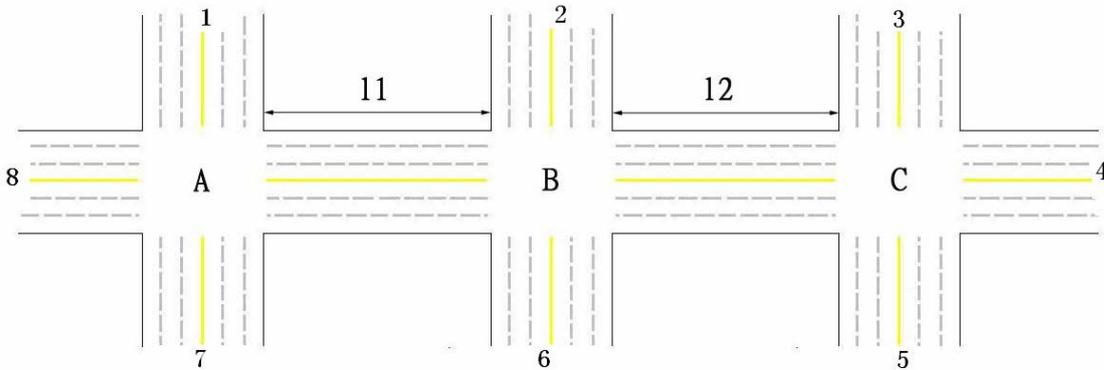


图 5.2

为了避免讨论各交叉口之间的关系，我们设图中标号的 8 个入口处车辆流入量为 U_i (i 是 1 到 8 的整数)，设 8 个口的车辆流出量为 X_i (i 是 1-8 的整数) 那么只要满足下列要求，即求得

$$\min \sum_{i=1}^8 \int t \cdot X_i(t) dt \quad (5.6)$$

就能等效的认为总等待时间最小。由于 U_i 为已知， X_i 由各路口的绿信比和周期以及 U_i 确定，所以可通过确定个路口的绿信比和周期实现上述目标。因竞赛时间有限，此算法将留在将来进一步讨论。

七、给交通管理部门的建议

城市交通信号控制极大的影响着道路交通系统运行效率和安全性。根据不同时间、地点交通流量的实时大小，对交通信号进行科学合理的实时配置，将有助于交通系统整体效能的提高。这里我们基于本文所建立的交通信号实时控制数学模型，提出一些建议：

- 1) 在主要交叉路口可设立车辆计数器，起到监测某段时间内流过车流总量的作用，若用已知的车流总量除以时间，就能得到流量的预测值，进而就能利用本文中的模型进行实时控制。
- 2) 当然，强烈建议交通管理部门对路口全天各时段车流量进行测量与统计，绘制不同时段车辆的经验分布，待车辆计数器测得车辆数并求出实时流量后，参照此经验分布确定此随机流量的可信度，或按照经验分布对已有数据进行修正。
- 3) 在进行实时控制的情况下，要求主干道与支路之间的饱和流量比差距尽量不要太大，一般建议在 2 倍以内，否则支路将始终被配以较大的绿信比，在高峰时间段，整个模型的计算速度会相当缓慢，这将会影响实时性。

八、程序附录

1) 软件环境

Matlab v7.1, 不同版本 Matlab 之间可能出现不兼容情况。

2) 系统文件部分^[5]

fbfun4.m

格式: `fun=fbfun4(x)`

输入: `x`, 长度为 5 的向量, `x(1)-x(4)` 分别表示交通信号第一相位至第四相位的有效绿信比, `x(5)` 表示交通信号的周期。

输出: 单位时间内所有车辆的总延误时间。

介绍: 在十字路口 4 相位的情况下, 使用需要优化的 5 个变量 (4 个相位的有效绿信比和周期) 作为输入来得到需要优化的总延误时间的函数。由于在寻优过程中仅仅考虑总延误时间与该 5 个变量之间的关系, 两条交叉路的流量和饱和流量的值作为全局函数在程序内部申明, 在程序外部赋值。

confun4.m

格式: `[c,ceq]=confun4(x)`

输入: `x`, 同 `fbfun4.m` 中 `x` 的定义。

输出: `c`, `ceq`, 非线性约束的条件。

介绍: 该函数对 `x` 的非线性约束进行定义。

fb4.m

格式: `[x fval]=fb4(paraz)`

输入: `paraz`, 长度为 4 的向量, 设交叉的两条道路平均流量不一致, 则 `paraz(1)` 表示主干道的流量, `paraz(2)` 表示支路的流量, `paraz(3)` 表示主干道的饱和流量, `paraz(4)` 表示支路的饱和流量。

输出: `x`, 同 `fbfun4.m` 中 `x` 的定义。 `fval`, 在 `paraz` 给定的条件下根据 `x` 而计算出的总延误时间的值。

介绍: 根据两条道路的流量输入以及饱和流量的定义来配置 `x`, 目标是使在满足一定约束条件的前提下, `fval` 总延误时间最短。约束条件在程序内部以及 `confun4.m` 里定义。该程序在 `x` 的各种组合情况中寻找最优解, 并根据输入的流量和饱和流量情况以及 `x` 的最优解来求出该控制模式的总延误时间。用户文件 `no1.m` 有该函数的范例。

fbt4.m

格式: `[x fval]=fbt4(paraz)`

输入: 同 `fb4.m`

输出: 同 `fb4.m`

介绍: 该函数与 `fb4` 基本类似, 区别在于它是在信号周期固定的条件下对 `x` 进行最优解的配置 (也就是说只考虑 `x(1)-x(4)` 的组合, 而 `x(5)` 是确定的)。该函数只用于周期固定的场合, 实现对各个有效绿信比进行最优配置。

linecon.m

格式: `[xm1 fm1 xm2 fm2 phase]=linecon(paraz,l)`

介绍: 对 3 个交叉路口组成的系统进行线控制。该函数给出了两种算法对 3 个叉路口的

有效绿信比和周期进行设置，并比较了这两种算法的控制效果。详见用户文件 no2.m 介绍。

spacecon.m

格式: [xm1 fm1 xm2 fm2 phase]=spacecon(paraz,l)

介绍: 对 6 个交叉路口组成的系统进行面控制。该函数同样给出两种算法对 6 个交叉路口的有效绿信比和周期进行设置，并比较了这两种算法的控制效果。详见用户文件 no3.m 介绍。

poisson_fb4.m

格式: [xps1 fps1 xps2 fps2 xps3 fps3]=poisson_fb4(paraz)

介绍: 在实际流量输入是随机流量而非流量平均值的情况下，该函数产生服从泊松分布的随机流量，采用三种方法对单交叉路口的控制信号进行配置，并分析各种控制方法对随机流量的控制效果。详见用户文件 no4.m 介绍。

poisson_linecon.m

格式: [xps fps phaseps]=poisson_linecon(paraz,l)

介绍: 在实际流量输入是随机流量的情况下，该函数产生 3 个交叉路口服从泊松分布的随机流量，采用变周期、变有效绿信比的算法对这 3 个交叉路口进行线控制，并返回控制结果和各个交叉路口的相位差。详见用户文件 no5.m 介绍。

poisson_spacecon.m

格式: function [xps fps phaseps]=poisson_spacecon(paraz,l)

介绍: 在实际流量输入是随机流量的情况下，该函数产生 6 个交叉路口服从泊松分布的随机流量，采用变周期、变有效绿信比的算法对这 6 个交叉路口进行面控制，并返回控制结果和各个交叉路口的相位差。详见用户文件 no6.m 介绍。

fbfun3.m、confun3.m、fb3.m、fbt3.m.....

格式与功能与前面介绍的 fbfun4、confun4 等文件相同，区别是这些文件适用于丁路口 3 相位的场合。

用户文件部分:

main.m

格式: 在命令窗口输入 main 并回车，在输入提示符后可输入 1-6 中的某个数字来调用 no1、no2、no3、no4、no5、no6 这六个程序。

no1.m

格式: 可在 main 命令中输入 1 调用，也可直接在命令窗口输入 no1 并回车。

介绍: 基于平均流量的点控制，fb4.m 和 fb3.m 的范例。输入为主干道流量、支路流量、主干道饱和流量、支路饱和流量，该范例取[1300 500 8000 4000]，输出 x1 表示十字路口 4 个相位和周期的最优组合，fval1 表示根据 x1 的控制计算出的总延迟时间，xc1 表

示基于经验公式的 4 个相位和周期的组合， $fc1$ 表示根据 $xc1$ 的控制计算出的总延迟时间， $x2$ 、 $fval2$ 、 $xc2$ 、 $fc2$ 表示的是丁字路口的情况。由于 $x2$ 和 $xc2$ 只有 3 个相位，为了保持 x 解的一致性，取 $x(4)=0$ ， $x(5)$ 仍表示周期。

no2.m

格式：可在 `main` 命令中输入 2 调用，也可直接在命令窗口输入 `no2` 并回车。

介绍：基于平均流量的线控制，`linecon.m` 的范例。输入为 3 个叉路口的主干道流量、支路流量、主干道饱和流量、支路饱和流量，该范例取[1500 500 8000 4000;1200 350 8000 4000;1300 500 8000 4000]，以及两个叉路口之间的距离，该范例取[300 200]，输出 $xm1$ 和 $xm2$ 分别表示传统算法和改良算法计算出的 4 个相位和周期的最优组合， $fm1$ 和 $fm2$ 分别表示 $xm1$ 和 $xm2$ 配置的总延迟时间， $phase$ 表示各个叉路口相对第一个叉路口的相位差， $sumnew1$ 和 $sumnew2$ 表示基于两种算法的 3 个叉路口的总延迟时间。

no3.m

格式：可在 `main` 命令中输入 3 调用，也可直接在命令窗口输入 `no3` 并回车。

介绍：基于平均流量的面控制，`spacecon.m` 的范例。输入为 6 个叉路口的主干道流量、支路流量、主干道饱和流量、支路饱和流量，该范例取[1500 500 8000 4000;1500 400 8000 4000;1400 500 8000 4000;1200 350 8000 4000;1300 500 8000 4000;1300 300 8000 4000]，以及各个叉路口之间的距离，该范例取[300 200 150 200 300 150]，输出 $xm1$ 、 $xm2$ 、 $fm1$ 、 $fm2$ 、 $phase$ 、 $sumnew1$ 、 $sumnew2$ 的含义与 `no2` 类似。

no4.m

格式：可在 `main` 命令中输入 4 调用，也可直接在命令窗口输入 `no4` 并回车。

介绍：基于随机流量的实时点控制，`poisson_fb4.m` 的范例。因为都是对单点的控制，输入为主干道流量、支路流量、主干道饱和流量、支路饱和流量，该范例取[1300 500 8000 4000]，但它仅代表主干道流量和支路流量的预估计值。输出 $xps1$ 表示流量输入是平均值情况下计算出的 x 最优解， $xps2$ 表示流量输入是随机值且要求周期固定的情况下计算出的 x 最优解， $xps3$ 表示流量输入是随机值情况下计算出的 x 最优解，有效绿信比和周期都是实时变化的。 $fps1$ ， $fps2$ ， $fps3$ 为这三种控制算法得出的总延误时间。经过 10 个周期的循环后，求这三种算法延误时间的累加和，用 $sum1$ 、 $sum2$ 、 $sum3$ 表示。

no5.m

格式：可在 `main` 命令中输入 5 调用，也可直接在命令窗口输入 `no5` 并回车。

介绍：基于随机流量的实时线控制，`poisson_linecon.m` 的范例。输入为 3 个叉路口的主干道流量、支路流量、主干道饱和流量、支路饱和流量，该范例取[1500 500 8000 4000;1200 350 8000 4000;1300 500 8000 4000]，以及两个叉路口之间的距离，该范例取[300 200]，但它仅代表主干道流量和支路流量的预估计值。输出 xps 每 5×3 部分表示 3 个叉路口的 x 最优解。 fps 每行表示 3 个叉路口的延迟时间。 $phaseps$ 每行表示各个交叉路口的相位差（以第一个叉路口为 0 来定义）。由于数据处理量相对较大，循环 5 个周期。

no6.m

格式：可在 `main` 命令中输入 6 调用，也可直接在命令窗口输入 `no6` 并回车。

介绍：基于随机流量的实时面控制，`poisson_spacecon.m` 的范例。输入为 6 个叉路口的

主干道流量、支路流量、主干道饱和流量、支路饱和流量，该范例取[1500 500 8000 4000;1500 400 8000 4000;1400 500 8000 4000;1200 350 8000 4000;1300 500 8000 4000;1300 300 8000 4000]，以及各个叉路口之间的距离，该范例取[300 200 150 200 300 150]，但它仅仅代表主干道流量和支路流量的预估计值。输出 `xps` 每 5×6 部分表示 6 个叉路口的 `x` 最优解。`fps` 每行表示 6 个叉路口的延迟时间。`phaseps` 每行表示各个交叉路口的相位差（以第一个叉路口为 0 来定义）。由于数据处理量相对较大，循环 5 个周期。

`no5.m` 和 `no6.m` 由于数据处理量的缘故，可能需要耐心的等待时间。

六、参考文献

- [1] 吴明晖，城市交通信号优化及仿真研究，硕士学位论文，P1-2，P56，2005
- [2] 林元烈，应用随机过程，北京：清华大学出版社，2002
- [3] 李江，交通工程学，北京：人民交通出版社，2002
- [4] 杨佩昆，交通管理与控制，北京：人民交通出版社，2003
- [5] 陈杰，Matlab 宝典，北京：电子工业出版社，2007