**第一次作业情况汇总**

1. 作业完成总体情况

根据完成的作业来看，对于习题一，多数同学们能够根据运用穷举遍历的思想，正确使用matlab编程语言编写相关脚本程序，得到满足题意的两组结果；有的同学学习并使用了perms（）函数，也能得到相同的正确结果；少数同学对程序耗时问题做了分析与优化，很好。对于习题二，多数同学都能根据题目条件完成建模，做出言之有理的分析。大家的作业都很不错，但在这里还是想提一些问题：

关于作业格式：

本次作业要求上传代码与分析文档，多数同学做的很好，但仍存在下列问题：有的同学将代码附在了对应的word文档中，而没有单独上传.m文件；有的同学只做了一道题目；有的同学没有按照分组交作业，希望之后能改进这些问题。

有27位同学没有及时提交作业，希望能及时补上。

关于习题一：

（1）同学们都能将正确的两组结果输出到命令行窗口中，但一部分没有经过处理，使得结果呈现的没那么直观。当然，大部分同学做了简单的处理，使命令行窗口内的输出形如“ 1738\*4=6952 1963\*4=7852 ”，很好。

（2）少有同学考虑了将得到的结果保存到工作区变量中的问题，数据保存还是挺重要的。

关于习题二：

（1）同学们都能根据自己的理解，对求解最优化的红绿灯时间做出相应的解答，给出答案，一些同学还给出不同的模型进行分析比较，还有一些同学把计算结果通过图像和表格进行直观的分析，值得学习。

（2）只有一部分同学不仅写出了解析思路，而且对计算结果进行分析，思考模型的不足之处，是一个较为完整的思考体系。

二、作业展示

这次的作业大家都完成的很好，有很多经过独立思考之后完成的优秀作业，但是由于有一些思路上的重复以及时间关系，我只选取了这些优秀作业中的一部分，对其中一些点给出了非常简单的点评，希望大家能理解。

习题一

1.董子铭 2017301020192

通过先定义了包含1至9的数组，然后再把这些数按顺序放进9个变量中，再对这9个变量进行适当的加减乘除使其变为两个4位数和一个1位数，通过randperm语句打乱数组的顺序，放进while语句中循环，只要不满足题目条件他就会打乱数组然后重新按上述流程测试，直至满足条件后停止循环并输出这两个4位数和一个1位数.

代码：

x=[1,2,3,4,5,6,7,8,9]

x1=x(1,1),x2=x(1,2),x3=x(1,3),x4=x(1,4),x5=x(1,5),x6=x(1,6),x7=x(1,7),x8=x(1,8),x9=x(1,9)

y1=x1\*1000+x2\*100+x3\*10+x4

y2=x6\*1000+x7\*100+x8\*10+x9

y3=x5

while y1\*y3~=y2

z=randperm(size(x,2))

x=z

x1=x(1,1),x2=x(1,2),x3=x(1,3),x4=x(1,4),x5=x(1,5),x6=x(1,6),x7=x(1,7),x8=x(1,8),x9=x(1,9)

y1=x1\*1000+x2\*100+x3\*10+x4

y2=x6\*1000+x7\*100+x8\*10+x9,y3=x5;end, y1, y2, y3

结果截图：



优点：思路新颖，通过randperm语句打乱数组的顺序，放进while语句中循环，输出满足条件的数组，是一个很独特的解题方法。

2.何泽楷 2017301020154

## 问题重述：

将1~9这就九个数不重复不遗漏的填入 □□□□×□=□□□□ 中, 使得等式成立.

## 解题思路

优化思路是将计算**向量化**

利用 MATLAB 强大的矩阵计算能力, 生成1~9的全排列矩阵后直接计算找出正确结果(直接计算指用全排列矩阵第一行乘1000加第二列乘100…算出所有结果判断是否满足等式), 见代码 ex01.m

%% Solution for Exercise 1  
% 解算思路：利用MATLAB自带的perms函数列出 1:9 的所有排列, 然后从这些排列  
% 中直接找出答案  
function out = ex01()  
%EX01 Finding solution for Problem 01  
 a = perms(1:9);  
 out = a((a(:, 1)\*1000 + a(:, 2) \* 100 + a(:, 3) \* 10 + a(:, 4)) .\* a(:, 5) == (a(:, 6)\*1000 + a(:, 7) \* 100 + a(:, 8) \* 10 + a(:, 9)), :);  
end

运行示例:

>> fprintf("%d%d%d%d \* %d = %d%d%d%d\n", ex01()');  
1963 \* 4 = 7852  
1738 \* 4 = 6952

## 算法效率和有效性分析

* 虽然进行了可能没有必要进行的计算, 但实际计算时间很短(用timeit测得小于0.05秒).
* 此算法可以保证计算出所有符合题目条件的结果.

优点：学以致用，不采用循环嵌套的方法，而是直接生成1-9的全排矩阵，对矩阵中的数据进行计算，判断是否符合要求，代码简单明了，计算时间相对少。

3.孙瑞祺 学号：2018302020073

解题分析：

本题目的难点在于如何使得计算过程中所用到的数字不重复，我最开始的思路是在循环结构中将每一个数字与其他数字进行比较，若不相同则判断等式是否相等,然而该算法较为复杂且不够简洁，所以我采取了另外一种思路，一开始生成一个元素相互独立且在1到9之间的九维向量x，然后用向量前四个元素表示第一个数字，第五个元素表示第二个数字，后四个元素表示第三个数字，用循环语句持续进行判定，若等式成立则输出向量x.

解题过程：

在正式的求解之前，我先用一段代码期望能够先把能使等式成立的向量x找出来，于是用到了以下代码：

while(1)

x=randperm(9);

if (x(1)\*1000+x(2)\*100+x(3)\*10+x(4))\*x(5)==(x(6)\*1000+x(7)\*100+x(8)\*10+x(9))

x

End

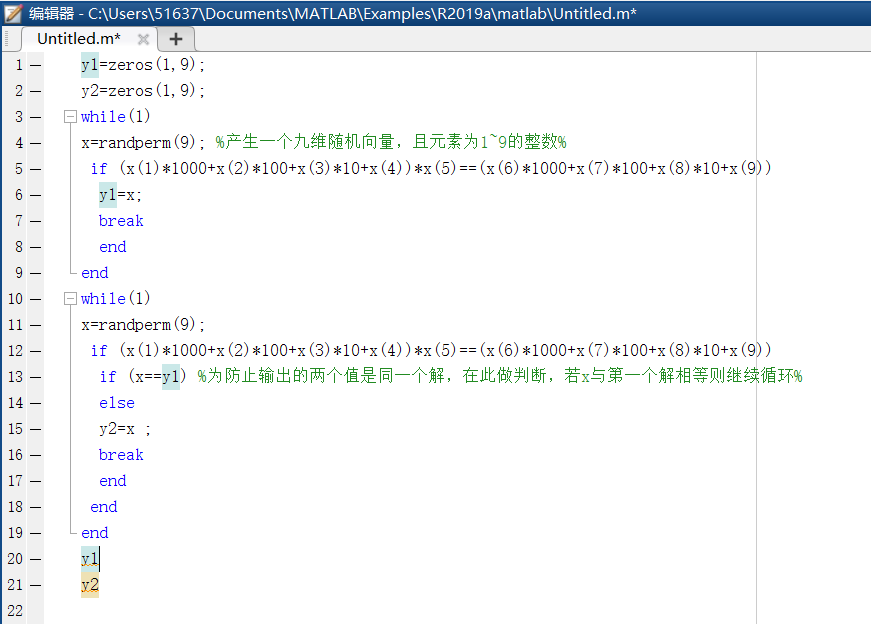
end

通过一个死循环，找到了该题目的两个解：

1738\*4=6952

1963\*4=7852

在知道了本题目只有两个解之后，为了使输出值更加简洁直观我对代码进行了修改，得到了最终版本。



其中用到的比较重要的一个函数是randperm，该函数能够生成一个随机的九维向量，且元素为1~9的不重复的整数值，很好的解决了九个数字不能重复的问题，另外为了防止输出的两个解为相同值，在第二个循环里多加了一次判断，保证第二次循环输出的值与第一次循环不同。

优点：通过对代码进行改进，得到了更好的方法。

习题二

1.何泽楷 2017301020154

## 分析

车道1仅直行, 车道4仅右转, 不影响其它车道, 故车道1, 4不设红绿灯. 车道2左转影响车道3, 应在2,3车道设置互斥的红绿灯.

## 解题思路

优化目标为“通行效率”. 设红绿灯周期为, 在一个红绿灯周期中, 车道2的绿灯时间为. 定义度量路口通行效率的函数如下：

其意义是平均每秒通过路口的车辆数(只考察设置了红绿灯的2,3车道). 为保证路口通过的公平性, 人为规定路口2绿灯时间占红绿灯周期的比例不超过, 不低于, 则优化目标为

等价问题为

## 这是一带约束条件的优化问题, 可用 MATLAB 中的 fmincon 函数求解(需要 Optimization To代码

定义优化目标函数ex02(保存在文件ex02.m中), 代码如下:

function [outputArg1] = ex02(X)  
%EX02 function to optimize for Exercise 02  
% Average cars per seconds, negate the result to convert maximun optim.  
% problem to minimum optimization problem  
outputArg1 = -(0.2 .\* X(:,1) ./ X(:,2) + 0.3 .\* (1 - X(:,1)./X(:,2)));  
end

在脚本文件ex02\_script.m执行优化函数并显示结果, 代码如下:

%% ex02\_script  
% Script for finding optimal result for problem in Exercise 2  
clc; clear ex02;  
A = [-1,0.4;1,-0.6]; % linear contraint matrix for 0.4t<=t\_g<=0.6t, t > 0  
% start with t\_g=15 secs and t=30 secs  
[x, fval] = fmincon('ex02', [15, 30], A, zeros(2,1), ...  
 [], [], [0, 0], []);  
fprintf("\ngreen light time: %gs, traffic light peroid: %gs\n", x);  
fprintf("Optimal cars per seconds: %g\n", -ex02(x));

此处使用了 [x, fval] = fmincon(FUN,X0,A,B,Aeq,Beq,LB,UB) 函数接口, 表示对给定函数FUN从X0开始在A\*X <= B, Aeq\*X = Beq的约束下, 在LB <= X <= UB的范围内函数的最小值, 返回最小值点x和最小值fval.

olbox).

## 运行示例

输入

>> ex02\_script

得到运行结果

Local minimum found that satisfies the constraints.  
  
Optimization completed because the objective function is non-decreasing in   
feasible directions, to within the value of the optimality tolerance,  
and constraints are satisfied to within the value of the constraint tolerance.  
  
<stopping criteria details>  
  
green light time: 12.5363s, Traffic light peroid: 31.3408s  
Optimal cars per seconds: 0.26

结果表明对车道2设置绿灯的最优时长为, 总的红绿灯周期为, 在此情况下路口通行效率为每秒辆.

## 模型的问题

该模型中各变量是车道车流量的线性函数, 使用线性优化得到的结果容易出现“赢者通吃”的现象, 即车流量最大的车道通行时间最长, 这样做在实际中是否合理需要进一步讨论. 另外此模型可能不能很好地反映其它实际因素, 如车流不均匀性, 同时通过的车之间的相互影响, 车辆在绿灯后需要一定时间才能启动等, 仍需进一步改进.

优点：定义度量路口通行效率的函数，并且对结果进行优化，最后分析反思了模型的不足之处。

2.朱若溪 2018302020231

解题分析：由题目中的示意图可知，1车道和其他车道互不影响，故不用考虑；由于方向相互冲突，2和3不能同时绿灯；由于5车道的承载力有限，2和4也不能同时绿灯。因此可以认为3和4共用一个信号灯，而2和它们的信号刚好相反。

建模过程：

为了简化问题，做出的基本假设是：车流均匀通过，且通过路口时的速度不变（即驶入速度等于驶出速度），忽略反应时间和转弯的时间。在这个前提下，一直都有汽车驶入但汽车不能一直驶出，所以驶来的车一定多于驶出的车，车队长度一定是净增加的。我们的目标是使几个车道等待的车队的长度比较均衡，或者说尽量推迟过长车队的产生。

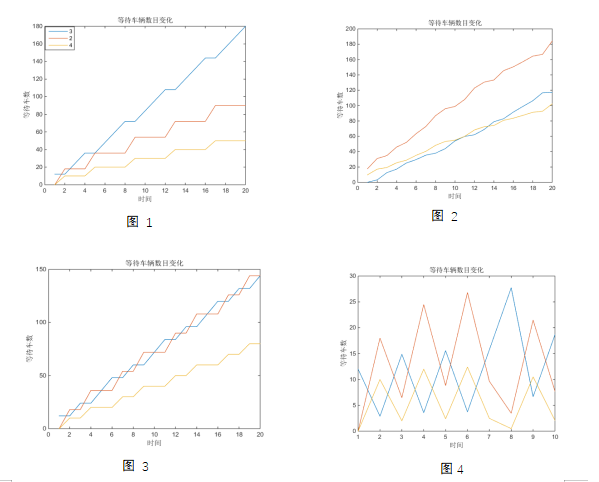
在第一个模型里，设一个周期为一分钟，决策变量是第2车道一个周期内的绿灯时间t（0<t<1）。引入三个偏好因数a, b, c, 且a+b+c=1。 目标函数f是三个偏好因数分别乘以各自车道的等待车辆数再相加。根据f的表达式f=a\*（12\*(1-t)+wait2)+b\*(18\*t+wait3)+c\*(10\*t+wait4) (其中wait是本轮周期初始时车道中的等待车辆数)，f是t的一次函数且t的系数与a, b, c有关。因此当t的系数不为零时，若要f最小，t只能取0或1。引入反馈调节，每过一个周期后根据各个车道最新的等待车辆数调整a, b, c，使偏好因数与车队长度成比例。编写程序进行20个周期，结果如附图一。最终队长为180，90，50。每个周期内变量t的值分别为：0，1，0，0，1，0，0，0，1，0，0，0，1，0 ， 0，0，1，0，0，0， 即稳定后2车道与3、4车道绿灯时间的比是1/3。

作为对比，给出当t随机取值时车队长度的变化，见附图二。其最终队长为179.6662，120.2225，99.8146

对比可知，用模型一可以有效控制其中两个车道的队长增长，但是第3车道队长增长过快，没有达到调节各个车道使之平衡的目的。所以在第二个模型中，调整了偏好因数的计算方法（每周期后最长车道的偏好因数置为1，其他为0），控制了最大队长。得到的结果见图三。每周期变量t的值为：0，1，0，1，0，0，1，0，1，0，0，1，0，1，0，0，1，0，1，0，成周期性变化。

在前面两个模型中，没有考虑路口处车流量的变化，导致得出的结果与实际差别较大。实际上由于路口处车流较密，车流量会增大。查资料得知路口处车流量约为50辆每分钟。在第三个模型中，根据这个值调整了车队长度的递推关系式，进行十个周期。这样得到结果更接近实际，折线图见图四。变量t值为 0，1，0，1，0，1，0，0，1，0

下表是等待车辆的数目随时间的变化图：



优点：建模十分认真，通过对模型结果的分析，不断对模型进行修改，得到更加全面的结果，并且查阅资料使得模型更加贴近现实。

3.\*\*\* \*\*\*\*(鉴于这位同学真诚的不想把他的作业拿出来，就匿名吧)

最优化方案考虑了以下两种情况：

1.一个周期内2、3、4车道滞留车辆总数最少。

2.一个周期内2、3、4车道车辆滞留时间总和最小（或平均滞留时间最小）。

且t值都应取最小值。

详细如下：

1. 设一个周期内2、3、4车道滞留车辆总数为z

z= t/5-x/4+ x/2-t/3+ x/3-t/30

=7x/12+t/30

满足条件：①红灯时滞留数需大于等于0，否则说明绿灯时已无需要通过的车辆，明显不合理。

得到t/5-x/4>=0

x/2-t/3>=0

x/3-t/30>=0

②周期t大于等于1分钟（否则感觉不太对……）

即t>=60

③每分钟在2车道绿灯时，至少能通过一半车辆（感觉这样行车效率较高）

得到 x>=4\*6=24

故化为线性规划问题：

min z=7x/12+t/30

满足：x/4-t/5<=0

-x/2+t/3<=0

x>=24

t>=60

代码如下：

f=[7/12;1/30];

A=[1/4 -1/5;-1/2 1/3];

b=[0;0];

lb=[24;60];

[x,fval,exitflag,output]=linprog(f,A,b,[],[],lb)

2. 设一个周期内2、3、4车道车辆滞留时间总和为Z

Z=y\*(t/5-x/4)+x\*(x/2-t/3+x/3-t/30) （红灯时间乘滞留数量）

=t^2/5+13x^2/12-49xt/60

满足条件与第一种最优化处理相同。

化为二次规划问题：

min Z=t^2/5+13x^2/12-49xt/60

满足：x/4-t/5<=0

-x/2+t/3<=0

x>=24

t>=60

代码如下：

H=[2/5 -49/60;-49/60,13/6];

f=[0,0];

A=[1/4 -1/5;-1/2 1/3];

b=[0;0];

lb=[24;60];

[x,fval]=quadprog(H,f,A,b,[],[],lb)

模型待修正地方：

1.实际每车道车辆流动速度不是均匀的，我认为这是模型存在最大的问题。

2.初始条件设2、3、4车道每过一辆车分别需要4、3、2秒不是很合理。

3.在红绿灯交替的临界点，可能出现变灯时有车通过或未完全通过的情况。

4.驾驶者在面临红绿灯时存在反应时间，不过可以粗糙地认为取平均并算在了过一辆车所需时间中。

优点：从“滞留车辆总数最少”和“平均滞留时间最小”两个视角看待最优化问题，分别进行建模与分析，并且指出模型待修正的地方，思路很完整。

4.吴杰雄 2018302020229

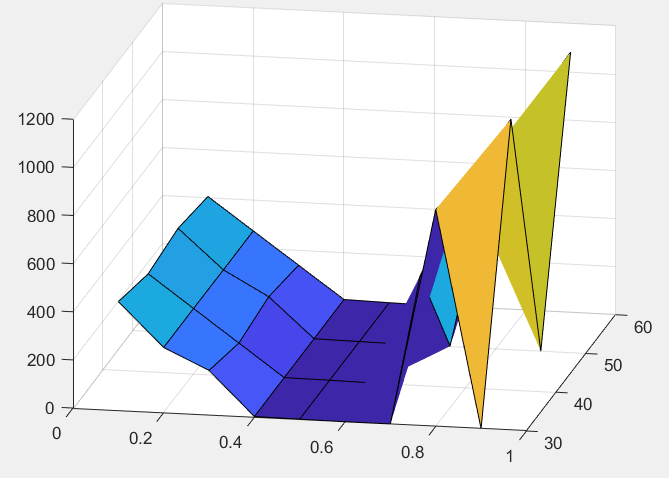
题目分析

本题首先设定了符合实际情况的设定，即第一道行不受限制。同时由于第五道是单行道，可以得到第2/3/4道的关系是，当第二道红灯时，第三和第四道绿灯；第二道绿灯时，第三四道红灯。

本题在实际操作时，首先是以一分钟后路口里的剩余车辆，但是实际计算后发现使得路口里车辆为零的情况红绿灯时间分配范围很大，而且路口一段时间积累的车辆接近于零这一指标是保证不堵车的必要条件，这个指标对实际情况没有指导意义。故本次的指标为使得一个小时内通过路口车辆的平均等待时间，即一小时内每个车通过路口等待时间之和除以一个小时内通过路口的全部车辆数。之所以选择一个小时的时间，是考虑到一般一个交通情况的平均时间约为1小时。在算法实现中将时间做了离散处理，以1秒钟为基本单位，同时认为每个车道是每过一个固定的时间通过一辆车。同时设置了可调节变量-每个车道平均在绿灯时每秒钟可通过路口的车辆数，方便根据不同情况进行调节（路段的不同会影响车速）。

本题将对红绿灯设置了两个因素，一个是第二道红绿灯一轮变化经过的时长，分别设置了30/40/50/60秒四种情况；另外一个因素是第二道红绿灯变化时，绿灯时间占总时间的比例，分别设置了0.1/0.2/0.3/0.4/0.5/0.6/0.7/0.8/0.9共九种变量。两种变量共组成了36种红绿灯分配状态。然后带入了当三个车道绿灯时每秒钟可以通过0.5辆车时的特例进行了计算分析作为算法可行性的验证。

带入特例后的结果为：

Z轴为每辆车平均等待时间

把结果用表格表示

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| 30 | 451.4946 | 270.8983 | 184.7313 | 2.940833 | 1.595417 | 0 | 0.698333 | 901.8983 | 0.249583 |
| 40 | 406.3483 | 271.7579 | 137.6408 | 3.547917 | 2.240833 | 0 | 456.0954 | 857.9483 | 1125.55 |
| 50 | 434.33 | 271.1654 | 169.2433 | 4.118333 | 1.820833 | 0 | 1.890833 | 832.0583 | 0.468333 |
| 60 | 405.6733 | 272.0658 | 139.1283 | 6.21 | 2.410417 | 0 | 459.5408 | 815.6454 | 1082.048 |

发现在这种情况中当第二道绿灯时间与红灯时间为1:1左右时的每辆车平均等待时间较小，而每轮红绿灯循环时间影响较小。当比例为3:2时平均等待时间为零，可能是因为算法在由于将时间以1秒钟为最小单位进行离散化处理造成的误差。但不影响这种分配方法在实际情况下每辆车平均等待时间较小的性质，即误差只会在某些特殊时间点产生较小误差。

本函数在实际运用于不同情境时可以通过改变输入参数调节，计算最适合设置方法。

优点:通过图形和表格把结果展示出来，清晰明了。