

浙江大学第十九届 大学生数学建模竞赛

2021 年 5 月 12 日—5 月 19 日

编号

5891

题目

A

B/

(在所选题目上打勾)

	参赛队员 1	参赛队员 2	参赛队员 3
姓名	徐圣泽	周宇昕	郑天宇
学号	3190102721	3190102722	3190103039
院(系)	竺可桢学院	数学科学学院	信息与工程学院
专业	数学与应用数学	数学与应用数学	电子科学与技术
手机	18258729819	19817868216	13738801331
Email	1762927943@qq.com	1614803512@qq.com	1712073729@qq.com

浙江大学本科生院
浙江大学数学建模实践基地

摘要

此次数学建模我们要解决的是一个企业车间内流动员工的专班安排问题。车间根据每天的员工数量会开设相应专班班次，我们所要解决的就是根据员工信息和各种要求、目标给出该车间内的专班排班结果。

考虑到每个人每天安排专班是一个是否的判断问题，可以设为 0-1 变量。那么整个排班问题相当于规划中的指派问题，是 0-1 整数规划。可以依据规划理论建立基本的 0-1 整数规划模型，写出基本的约束条件和目标函数。

规划的基本约束条件为每个专班限定两名员工、每人每天至多一次专班和一次专班两人至少一人有驾驶资格，这些约束都可由变量表达式给出。而该规划的复杂之处在于它的目标。整个排班结果总共需要满足四个原则，按重要程度分别为每位员工当前工作天数与当前累计安排专班次数之比尽可能均衡、每个班次至少一名员工有专班经验、每名员工不同天内参与专班在种类上尽可能平均以及历史事故次数高员工需尽量与历史事故次数低的员工组合。要尽可能满足最多和最重要原则，则该问题是一个多目标规划问题。

为处理多目标规划问题，我们采用常用的线性加权法，将各目标乘上权重得到总目标函数，多目标问题从而转化为单目标问题。于是在规划中根据各目标要求，我们对已有员工数据进行预处理后可给出各目标函数的具体形式。为保证比例一致，对它们进行一定的调整，之后假定总和为 1 的各权重，可得到线性加权过后的总目标函数。

至此，整个多目标的 0-1 整数规划模型基本建立，需要给出解决该特定规划问题的算法。基于题设要求，我们给出的排班结果分为一天和多天，规划求解也可按一天与多天考虑。一天的 0-1 整数规划可采用枚举算法，时间复杂度为多项式级。多次重复利用该算法，可以得到特定员工数据表下的单日依次排班结果。在取定合理权重的目标函数下，结果为特定值。而对于多天的 0-1 整数规划，若仍采用枚举算法，时间复杂度为指数级，无法实现。此处采用基本的启发式算法——随机搜索算法对问题做简单近似，可以得到近似情况下较优的多天排班结果。通过理论上的比较说明，我们可以说明该算法是可靠的，其结果接近真实最优解，其排班接近真实最优排班。综上，为得到排班结果，我们采用了两种算法，两种算法在具体结果上存在一定差异，我们对其作了对比，分析了结果出现的原因和算法的优劣性。

根据题设要求，在第二问中定义了新的公平，为此需要修改原来最首要的子目标函数 f_0 。根据新定义，我们相应定义新的目标函数 f'_0 ，得到新目标函数 f' 。将 f' 作为目标，放入原有算法。可以得到变换了公平条件后的结果。将原有结果与新结果进行比较，可以分析比较两种公平，并以此为基础对整个“公平”的含义进行相关探讨。

在第三问中，我们要回答分配新员工是否要考虑专班安排要求这个问题。问题的答案显然为是，我们通过改变原有的新员工输入，变换初始条件代入算法从而得到不同结果。根据结果分析，我们相应可以说明怎样的新员工分配方案是最合理的。

最后，我们在论文末尾对之前的算法可行性进行了补充分析，主要包括权重选取的合理性和采用随机搜索算法的合理性说明。同时，考量算法，我们总结了算法的问题并给出了初步可改进的策略。至此，论文正文部分全部结束，在附录我们附上了算法实现的代码和其中部分排班结果的显式展现。

关键字： 规划模型，目标函数，枚举算法，随机搜索算法，公平条件

1 情景概述

企业的员工一般具有很大的流动性，此次问题的情景即是一个有流动员工的企业车间。车间会根据每天工作的员工数量安排具体的专班班次。每个专班限定两名员工，每名员工每天至多一次专班。每名员工有关于他工作天数、是否有驾驶资格、历史事故次数等的记录。专班安排有基本要求，也有许多需满足的原则。依据具体的员工数据表，在满足这些要求和原则的基础上，要给出相应的专班安排结果。

2 解决思路

2.1 基本假设

为了使模型更贴合实际，代码更容易实现，我们作以下假设：

1、模型中初始的总员工数量确定，在具体场景下不管员工是否仍在工作，总员工数量仍为初始给定值，始终保持不变；

2、仅考虑一个车间内的人员流动和分工安排，在该车间内工作人数大于 15 或小于 5 时默认不符合条件，不进行排班。

3、模型中目标涉及的“当前”是一个变化概念，在排每一天的具体班次时，“当前”都指暂定的排班日，当前工作天数和累计专班次数均截止到排班日。在连续数天情况下，对每一天都会做“当前”排班日处理，得到相关函数。但同时考虑不同“当前”的时效性，假设越靠近现在、越少天数所对应的函数拥有越高的优先级。

4、由于历史事故次数随时间的变化未知，默认在排班过程中每位员工的历史事故次数不变。没有具体历史事故次数值的员工默认该值取 0。

2.2 解决方向

该问题基本为规划问题中的排班（指派）问题，我们的解决方向是采用规划的方法。在具体规划中，我们用 0-1 变量代表是否排班，整个规划问题就是 0-1 整数规划问题。

规划问题由约束条件和目标组成。由题设，该问题中的专班安排需要满足多个原则，且原则间有重要顺序，这些原则即是规划所要实现的目标。则该规划也是多目标规划问题。解决多目标规划，我们采用线性加权法，对各子目标函数赋上权重作为最终总目标函数。

第一个问题首先要求建立模型给出一般情形的专班安排算法，即要根据约束及目标建立完整的规划并尝试由算法实现排班。在具体算法实现中，我们注意到需要给出的是一天及连续数天的排班结果，其具体算法可以有所不同。对于一天，数据量有限，运算时间复杂度为多项式级，而本身待求量为 0-1 变量，可以采用枚举算法得到规划最优解。对于连续数天，问题的运算时间复杂度升级为指数级，枚举算法不再有效，我们的想法便是借助基本的启发式算法——随机搜索算法来实现连续数天的排班。

第二个问题要求探讨不同公平含义间的关系。在原有规划中，公平条件即是最首要的子目标函数。则在该内容中，通过更改公平条件对应的子目标函数表达式，代入已有排班算法进行比较，即可做出相应探究。

第三个问题要求探讨分配新员工时是否应考虑专班安排的要求。这一问题的答案显然是肯定的。为探究新员工对专班要求的影响，我们的想法是改变初始量，主要是改变初始量中新员工的组合，将改变过后的场景代入算法，根据运行结果得出相关结论。

3 规划理论

3.1 变量声明

在解决本题的过程中，由于所设变量数较多，于是我们将多次用到的变量和符号名记录在下表中：

表 1: 模型所需变量

变量	变量名
总天数	k
总员工数量	n
第 j 天的员工数量	n_j
到第 j 天已工作过的员工数量	m_j
工作天数	W_i
已工作天数	w_i
是否有驾驶资格的 0-1 变量	q_i
第 i 个人的历史事故次数	tt_i
衡量第 i 个人的历史事故次数的常值变量	t_i
第 i 个人已排具体班次	A_i, B_i, C_i, D_i
第 i 个人到第 j 天是否已有对应专班经验的 0-1 变量	$A'_{ij}, B'_{ij}, C'_{ij}, D'_{ij}$
第 j 天第 i 个人是否工作的 0-1 变量	x_{ij}
第 j 天是否安排第 i 个人专班的 0-1 变量	y_{ij}
第 i 个人到第 j 天的已工作天数	z_{ij}
第 j 天总共需要安排的具体班次对应人数	a_j, b_j, c_j, d_j
第 j 天是否安排第 i 个人负责具体专班的 0-1 变量	$a_{ij}, b_{ij}, c_{ij}, d_{ij}$
第 j 天第 i 个人的当前累计专班次数与当前工作天数之比	T_{ij}
第 j 天衡量员工当前累计安排专班次数与当前工作天数之比均衡的方差	δ_{1j}
连续数天情况下衡量 T_{ij} 均衡时第 j 天的比重	ω_j
衡量第 i 个人专班具体班次平均的方差	δ_{2i}
各目标函数在总目标中所占的权重	$\lambda_0, \lambda_1, \lambda_2, \lambda_3$

变量表中，各变量的下标 i 表示第 i 个员工， j 表示第 j 天。其中 $q_i, A'_{ij}, B'_{ij}, C'_{ij}, D'_{ij}, a_{ij}, b_{ij}, c_{ij}, d_{ij}, x_{ij}, y_{ij}$ 均为 0-1 变量。这些变量在表达是时取值为 1，否则取值为 0。

为解决问题，我们采用规划方法，所求得的结果即是最终的员工具体班次安排表，所求的目标变量即是上述的 $a_{ij}, b_{ij}, c_{ij}, d_{ij}$ 。其余所设变量大致分为以下几种：一种是根据模型所给的初始值设立的变量，这些变量依据所给表中数据、经过一定处理后成为模型中可代入的已知具体值的变量；另一种是带有目标变量、为使内部约束条件或目标函数清楚明晰而设立的辅助变量。以下是对这些变量和对应的规划约束条件、目标函数的具体说明，以此体现我们关于多天排班整个问题的模型建立。

3.2 数据预处理

在具体场景下，我们会得知待安排的总的员工数量 n 和一系列关于这些员工各项指标的变量，有些变量可按前变量声明直接读入，而有些变量需要进行相关预处理再存入。而同时，一些后续规

划所要用到的变量需要通过所给量存入相应的已知数值。

在场景所给值中，其已工作天数和历史事故次数的变量需要进行一定预处理再存入。对于已工作天数，那些未到岗员工在表中的呈现形式为 $+x$ ，在后续规划处要求下，我们希望这些代表未到岗的 $+x$ 存入的是 $-x$ ，即已工作天数 w_i 大于等于 0 时表示在开始排班首天已到岗，而小于 0 表示未到岗。 $1 - w_i$ 即代表该员工开始工作的那天。

此外，在专班安排次数中对于历史事故次数高低有大致划分，为后续规划需要，我们需将历史事故次数的绝对值 t_{ii} 转化为衡量其低中高的三级相对值，具体转换如下：

$$t_i = \begin{cases} 0, & t_{ii} = 0 \text{ 或 } 1; \\ 1, & t_{ii} = 2 \text{ 或 } 3; \\ 2, & t_{ii} > 3. \end{cases} \quad (1)$$

此外，在连续数天内，每一天的员工数量 n_j 和相应员工是否工作 x_{ij} 是根据所给量可取得的已知值。

$$x_{ij} = \begin{cases} 1, & 1 - w_i \leq j \leq W_i - w_i; \\ 0, & j < 1 - w_i \text{ 或 } j > W_i - w_i. \end{cases} \quad (2)$$

$$n_j = \sum_{i=1}^n x_{ij} \quad (3)$$

每一天工作的员工数量对应于每天需要开设的具体专班班次。该对应关系可由矩阵索引实现。设班次与工作人数的对应矩阵为 M ，行代表人数，列代表具体班次。

$$M = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 2 & 2 & 2 & 0 \\ 2 & 2 & 2 & 0 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \end{pmatrix} \quad (4)$$

由此得到对应的第 j 天需要安排 A、B、C、D 具体班次人数为 a_j 、 b_j 、 c_j 、 d_j ，

$$a_j = M(n_j, 1), b_j = M(n_j, 2), c_j = M(n_j, 3), d_j = M(n_j, 4) (n_j \leq 15) \quad (5)$$

$$a_j = b_j = c_j = d_j = 0 (n_j \geq 16) \quad (6)$$

此外，之后求一些均值时要用到第 i 个人到第 j 天的已工作天数 z_{ij} 和到第 j 天已工作过的员

工数量 m_j ，这也是已知量，需借助自定义的布尔函数 $isZero$ 得到。

$$z_{ij} = isZero(w_i + 1) \cdot w_i + \sum_{l=1}^j x_{il} \quad (7)$$

$$m_j = \sum_{i=1}^n isZero(z_{ij}) \quad (8)$$

其中在程序中可自定义的 $isZero$ 函数表达式为

$$isZero(x) = \begin{cases} 0, & x \leq 0; \\ 1, & x > 0. \end{cases} \quad (9)$$

3.3 约束条件

在题设条件下，所求目标量 a_{ij} 、 b_{ij} 、 c_{ij} 、 d_{ij} 需要满足一些基本的约束条件。每天总安排专班数固定，且每名员工每天至多安排一次专班，则对所有 $i(1 \leq i \leq n)$ 和 $j(1 \leq j \leq k)$ 满足

$$\sum_{i=1}^n a_{ij} = a_j, \sum_{i=1}^n b_{ij} = b_j, \sum_{i=1}^n c_{ij} = c_j, \sum_{i=1}^n d_{ij} = d_j \quad (10)$$

$$y_{ij} = a_{ij} + b_{ij} + c_{ij} + d_{ij} \leq x_{ij} \quad (11)$$

每天的每个专班需要安排两名员工，其中至少有一名有驾驶资格：

$$\sum_{i=1}^n a_{ij} \cdot q_i \geq \frac{a_j}{2} \quad (12)$$

$$\sum_{i=1}^n b_{ij} \cdot q_i \geq \frac{b_j}{2} \quad (13)$$

$$\sum_{i=1}^n c_{ij} \cdot q_i \geq \frac{c_j}{2} \quad (14)$$

$$\sum_{i=1}^n d_{ij} \cdot q_i \geq \frac{d_j}{2} \quad (15)$$

以上即为该问题最基本的约束条件，所求目标应该在这些约束条件基础上去尽量达成。

3.4 目标函数

我们的安排总体需要满足四个原则，相应地要设立四个子目标函数，每个函数乘上相应的权重之和即为总目标函数。

四个子目标函数按重要程度由高到低记为 f_0 、 f_1 、 f_2 、 f_3 ，对应权重为 λ_0 、 λ_1 、 λ_2 、 λ_3 ，最终目标函数

$$f = \lambda_0 f_0 + \lambda_1 f_1 + \lambda_2 f_2 + \lambda_3 f_3 \quad (16)$$

$$\lambda_0 + \lambda_1 + \lambda_2 + \lambda_3 = 1, \lambda_0 > \lambda_1 > \lambda_2 > \lambda_3 \quad (17)$$

最终目标为 $\min f$ 。

目标 0（公平原则） 每位员工当前累计专班次数与当前工作天数之比尽可能均衡

题设排班的首要目标为当前工作天数与当前累计安排专班次数之比尽可能均衡。由于累计安排专班次数很多时候会取到 0，此处做一个转换，以员工当前累计安排专班次数与当前工作天数的比值 T_{ij} 作为考虑的均衡量。到第 j 天的当前工作天数为 z_{ij} ，该值为整数，可能为 0。但我们可以注意到在分母为 0 时分子必为 0，则通过在分母里加一极小量 0.00001，可以在算式恒有意义的情况下与真实比值保持基本一致，具体形式如下：

$$T_{ij} = \frac{A_i + B_i + C_i + D_i + \sum_{l=1}^j (a_{il} + b_{il} + c_{il} + d_{il})}{z_{ij} + 0.00001} \quad (18)$$

均衡由方差体现，第 j 天的方差表达式如下：

$$\delta_{1j} = \sqrt{\frac{1}{m_j} \sum_{i=1}^n isZero(z_{ij}) (T_{ij} - \frac{\sum_{i=1}^n T_{ij}}{m_j})^2} \quad (19)$$

在累计 k 天过程中，每一天都会有一个方差取值。由于我们假设的当天根据时间有优先级，则 δ_{1j} 在最终的衡量目标中需要乘上相应的比重 ω_j ， $\omega_j (1 \leq j \leq k)$ 满足：

$$\omega_1 \geq \omega_2 \geq \dots \geq \omega_k \quad (20)$$

$$\sum_{j=1}^k \omega_j = 1 \quad (21)$$

此外，最终目标中我们要保持比例一致，此处以初始方差 δ_{10} 为比例参照值， δ_{10} 由初始比值 T_{i0} 和初始已工作人数 m_0 得到：

$$T_{i0} = \frac{A_i + B_i + C_i + D_i}{w_i + 0.00001} \quad (22)$$

$$m_0 = \sum_{i=1}^n isZero(w_i) \quad (23)$$

$$\delta_{10} = \sqrt{\frac{1}{m_0} \sum_{i=1}^n isZero(w_i) (T_{i0} - \frac{\sum_{i=1}^n T_{i0}}{m_0})^2} \quad (24)$$

由此得到目标 0 的子目标函数形式为：

$$f_0 = \frac{\sum_{j=1}^k \omega_j \delta_{1j}}{\delta_{10}} \quad (25)$$

为尽可能使最终解为优解，应使 f_0 的值尽可能小。

目标 1 每个班次尽量有一名有专班经验的员工

先用具体数学表达式表示专班经验：

$$\begin{aligned} A'_{ij} &= isZero(A_i + \sum_{l=1}^j a_{il}), B'_{ij} = isZero(B_i + \sum_{l=1}^j b_{il}), \\ C'_{ij} &= isZero(C_i + \sum_{l=1}^j c_{il}), D'_{ij} = isZero(D_i + \sum_{l=1}^j d_{il}) \end{aligned} \quad (26)$$

要求尽可能每个班次有一名有专班经验的员工，我们考虑对立情况，记 k 天内不符合要求的排班

数量为 U_k ,

$$\begin{aligned}
U_k = & \sum_{j=1}^k \left[\sum_{i_1=1}^{n-1} \sum_{i_2=i_1+1}^n a_{i_1j} a_{i_2j} (1 - A'_{i_1j})(1 - A'_{i_2j}) \right. \\
& + \sum_{i_1=1}^{n-1} \sum_{i_2=i_1+1}^n b_{i_1j} b_{i_2j} (1 - B'_{i_1j})(1 - B'_{i_2j}) \\
& + \sum_{i_1=1}^{n-1} \sum_{i_2=i_1+1}^n c_{i_1j} c_{i_2j} (1 - C'_{i_1j})(1 - C'_{i_2j}) \\
& \left. + \sum_{i_1=1}^{n-1} \sum_{i_2=i_1+1}^n d_{i_1j} d_{i_2j} (1 - D'_{i_1j})(1 - D'_{i_2j}) \right]
\end{aligned} \tag{27}$$

为保证各目标在最终目标中数量级尽可能接近, 此处以 U_k 除总排班数作为子目标函数 f_1 :

$$f_1 = \frac{U_k}{\sum_{j=1}^k (a_j + b_j + c_j + d_j)/2} \tag{28}$$

为尽可能使最终解为优解, 应使 f_1 的值尽可能小。

目标 2 每名员工不同专班种类尽可能平均

N_i 代表第 i 名员工安排不同专班次数的平均值:

$$N_i = \frac{(A_i + \sum_{j=1}^k a_{ij}) + (B_i + \sum_{j=1}^k b_{ij}) + (C_i + \sum_{j=1}^k c_{ij}) + (D_i + \sum_{j=1}^k d_{ij})}{4} \tag{29}$$

种类均衡由方差体现, 第 i 名员工的方差表达式如下:

$$\delta_{2i} = \sqrt{\frac{1}{4} [(A_i + \sum_{j=1}^k a_{ij} - N_i)^2 + (B_i + \sum_{j=1}^k b_{ij} - N_i)^2 + (C_i + \sum_{j=1}^k c_{ij} - N_i)^2 + (D_i + \sum_{j=1}^k d_{ij} - N_i)^2]} \tag{30}$$

与前面同理, 为保证各目标在最终目标中数量级尽可能接近, 我们还需计算得到初始均值:

$$N_{i0} = \frac{A_i + B_i + C_i + D_i}{4} \tag{31}$$

和初始方差:

$$\delta_{2i0} = \sqrt{\frac{1}{4} [(A_i - N_{i0})^2 + (B_i - N_{i0})^2 + (C_i - N_{i0})^2 + (D_i - N_{i0})^2]} \tag{32}$$

继而根据计算值得到 f_2 表达式:

$$f_2 = \frac{\sum_{i=1}^n \delta_{2i}}{\sum_{i=1}^n \delta_{2i0}} \tag{33}$$

为尽可能使最终解为优解, 应使 f_2 的值尽可能小。

目标 3 历史事故次数高的员工与历史事故次数低的员工配对组合

在此之前, 我们已经过历史事故次数进行了数据预处理, 定义新的目标函数 f_3 :

$$f_3 = \frac{\sum_{j=1}^k [\sum_{i_1=1}^{n-1} \sum_{i_2=i_1+1}^n (a_{i_1j} a_{i_2j} + b_{i_1j} b_{i_2j} + c_{i_1j} c_{i_2j} + d_{i_1j} d_{i_2j}) (t_{i_1} + t_{i_2})(t_{i_1} + t_{i_2} - 1)(t_{i_1} + t_{i_2} - 2)]}{3 \sum_{j=1}^k (a_j + b_j + c_j + d_j)} \tag{34}$$

在本表达式中, 若两人的历史事故次数均较高 (即 $t_{i_1} = t_{i_2} = 2$) 时, 或一人历史次数较高同时另一人历史事故次数不为较低 (即 $t_{i_1} = 2, t_{i_2} = 1$ 或 $t_{i_1} = 1, t_{i_2} = 2$) 时, 函数值不为 0, 其余情况函数值均为 0。

这也符合我们的预期, 希望使表达式的值尽可能小。

4 算法及结果

上述建立的规划模型对于一般情况都可以进行求解，在题设所给的具体场景下，我们给出规划实现的具体算法和相应结果。

针对题目要求，我们设计了两种算法，其本质是枚举算法和随机搜索算法。第一种算法用于进行单日依次排班；第二种算法在初始量的基础上进行连续多日的统筹排班。实际上，两种算法都可以解决另一种问题，但各有其优缺点，我们会在下文中阐述选择算法的合理性及优越性。

4.1 单日依次排班

在这部分中，我们利用枚举算法，根据已有量来一天天依次安排排班情况，此种情况下，计算第 x 天情况时第 $x - 1$ 天的排班情况作为已经计算得到的量是固定值。

由于在此种算法中，目标函数是针对每一个固定的天数而言的，因而对于多天目标函数而言本算法极有可能无法取到最优值，但是对于每一天而言，此算法可以尽可能地达到最优解。

我们在上面叙述的几个目标函数和各变量定义，对于这种算法而言，仍然是成立并且合理的。对于程序的算法而言，只需每次将程序的输出量再次作为已知量输入，重复运行即得到了每天的安排情况。

我们根据题目给出的各员工的信息，先处理得到下面有关各员工各天工作的具体情况的表格：

表 2: 各天工作情况

天数	工作人员编号	人数
1	1、2、3、4、5、6、7、8、9、10	10
2	1、2、4、5、6、7、8、9、10	9
3	2、4、5、6、8、10、11	7
4	2、4、5、6、8、10、11、12	8
5	2、4、5、8、10、11、12	7
6	10、11、12、13、14	5
7	10、11、12、13、14、15	6
8	10、11、12、13、14、15	6
9	10、12、13、14、15	5
10	12、14、15	3

可以发现题目中给出的情况是一种较为简单的情况，每天的工作人数都不超过 10 人，这意味着最多只需安排 A、B 两种专班。

我们根据上述数学理论和算法，利用 *matlab* 写出相关代码，运行程序得到了目标函数值和专班安排情况，其中目标函数值是衡量我们算法合理性和优越性的重要指标，我们在后文中会具体探讨这一点。

在本部分中，我们选取了合适的待定系数，使得定义目标函数的计算公式为 $f = 0.8f_0 + 0.16f_1 + 0.032f_2 + 0.008f_3$ （在后文算法可行性分析中，我们会对待定系数的确定进行讨论和说明）。

根据程序运行结果，我们得到每天的安排情况和对应的目标函数值，列在下表中。

表 3: 各天专班安排情况

天数	排班情况	f_0	f_1	f_2	f_3	f
1	A: 6、10 B: 3、8	0.3111	0	1.2791	0	0.2898112
2	A: 2、7 B: 5、6	0.296	0	1.3273	0.5	0.2832736
3	A: 2、6	0.2905	0	1.1582	0	0.2694624
4	A: 8、11 B: 4、10	0.26	0	1.0547	0	0.2417504
5	A: 4、12	0.2128	0	1.0561	0	0.2040352
6	A: 9、10	0.2156	0	0.9785	0	0.203792
7	A: 12、13	0.176	0	1.0202	0	0.1734464
8	A: 11、14	0.0798	0	1.0755	0	0.098256
9	A: 11、14	0.1245	0	1.1277	0	0.1356864

从安排表中可以得出在单日依次安排时目标函数 f 的值始终保持在 0.3 以下，靠近总的目标函数的最优下界值 0，规划在该算法下所得的确定解是优越的。

这样按照每日穷举得到的专班安排方式存在着一些考虑不够全面的问题，如在单日保障均衡时无法保障整体不同天数间的均衡、成员间排班不“公平”等，这些我们在后面会进行详细的讨论。

4.2 多日统筹排班

在本部分代码实现的过程中，我们发现如果在只根据初始量来安排所有天的专班的情况下，如果利用枚举法，对 n 个人安排 k 天排班时可能会出现 $O(n^{8k})$ 种数量级的情况，算法的时间复杂度是指数级的，因而利用穷举法去进行多日统筹排班显然是较难实现的。

因此在本部分的算法设计中，我们利用随机搜索算法来得到情况较优的排班情况，我们希望在所有可能的排班种类中，随机抽样求取最优解。

对于题目给定场景，我们在约束条件下通过随机生成不同数量可能排班的情况，来测试算法可能达到的效果（代码见附录）。下表即为运用该算法所得到的程序运行结果。在下表中，目标函数的计算公式为 $f = 0.8f_0 + 0.16f_1 + 0.032f_2 + 0.008f_3$ 。

表 4: 不同量数据下的目标函数值

不同数据量不同次运行结果	f_0	f_1	f_2	f_3	f
1w 随机数据第 1 次	0.6646	0	1.3640	0.1667	0.5767
10w 随机数据第 1 次	0.6226	0	1.3245	0.0833	0.5411
10w 随机数据第 2 次	0.6132	0	1.3132	0.5000	0.5366
10w 随机数据第 3 次	0.6362	0	1.3665	0.1667	0.5540
10w 随机数据第 4 次	0.5881	0	1.3166	0.2500	0.5146
10w 随机数据第 5 次	0.6062	0	1.3688	0.0833	0.5294
100w 随机数据第 1 次	0.5686	0	1.3915	0.3333	0.5021
100w 随机数据第 2 次	0.5995	0	1.2832	0.0833	0.5213
100w 随机数据第 3 次	0.5877	0	1.3757	0.4167	0.5175
100w 随机数据第 4 次	0.5685	0	1.3792	0.5000	0.5029
100w 随机数据第 5 次	0.5803	0	1.4210	0.7500	0.5157

根据程序运行的结果，我们发现随着数据量的增大，目标函数值向着更好即更小的方向发展。限于多天排班时 f_0 的值存在一定下界，总的目标函数会存在一定下界。在数据量为 100w 时，多次运行得到的目标函数值为 0.5021、0.5213、0.5175、0.5029、0.5157，它们的接近程度可以用差距最大值的比例体现： $(0.5251 - 0.5021)/0.5021 = 4.6\% < 5\%$ 。这说明在取 100w 随机数据时，所得的目标函数值已相当接近，而由于数据选取的随机性，它们的值也应当接近真实的最优解，它们可以被视作该规划问题的近似最优解。

下面对最优解进行大致估计。限于较大随机样本 100w 情况下已得近似最优解间的差距为 5%，而最优值小于近似最优解，则可得：

$$\text{最优解} \geq 0.5021 \times (1 - 5\%) = 0.4770 \quad (35)$$

0.4770 在该问题中即可被认为是一个下界，最优值与它的差距小于 5%，这说明该算法得出的结果与真实最优解误差小，相对精确。

而对于具体排班情况，我们选取 1w、10w、100w 数据下各一种的排班情况列在下表作为参考：

表 5: 不同量数据下的安排情况

天数	1w 随机数据排班情况	10w 随机数据排班情况	100w 随机数据排班情况
1	A: 6、10 B: 2、8	A: 6、8 B: 2、10	A: 2、6 B: 5、10
2	A: 1、2 B: 5、8	A: 1、4 B: 5、8	A: 6、8 B: 1、7
3	A: 6、11	A: 2、10	A: 2、10
4	A: 5、10 B: 4、12	A: 5、6 B: 8、11	A: 6、11 B: 4、10
5	A: 5、8	A: 5、12	A: 5、12
6	A: 10、11	A: 11、13	A: 10、11
7	A: 11、12	A: 11、14	A: 13、14
8	A: 11、14	A: 10、13	A: 13、15
9	A: 12、13	A: 13、15	A: 14、15

我们发现，虽然不同数据量下的安排情况不尽相同，但整体来看排班情况较为公平，对于每个人而言其专班次数和各种专班种类也较为均衡。

4.3 算法对比

可以发现，两种算法得到的结果具有比较大的差异，这是因为两种算法计算目标函数值的方式不同。主要的差异在于其中 f_0 和 f_2 的设立。

在单日依次排班时， f_0 只考虑了当天工作的人数，所有的比值在计算第 j 天的方差时只有 n_j 个相关样本，且采用的均是当日更新的量，因而在规划考虑时能使其尽量达到最小。而对于多日统筹排班， f_0 第 j 天的量考虑的是到第 j 天已工作过的人数，在计算第 j 天的方差时用到了 m_j 个相关样本。显然， $m_j \geq n_j$ ，且多日统筹安排时由于每位员工的工作情况不同，他们对应的比值必会有较大差异，尤其在老员工与新员工尤为明显。这也就解释了为什么第二种算法的 f_0 值普遍大于第一种算法。且第二种算法基于之前的规划，在 f_0 计算时对每一个当天 (j) 进行了 $(2k+1-2J)/k^2$ 的加权，这也使得数据有了一定变化。若改变 k 值，在 $k=1$ 时第二种算法所得的较优解中 $f_0 = 0.2332$ ，与第一种相差不大，此外 $k=3$ 时得 $f_0 = 0.3175$ ， $k=5$ 时 $f_0 = 0.4640$ ，具体结果也说明了第二种算法的 f_0 值会受天数影响且不断递增，但其结果依旧维持精确。

关于 f_2 ，同样第一种算法考虑的样本数是少于第二种的，其在每一个当天的值是会少于第二种多天安排的值。但可以发现单日依次排班时不同日间 f_2 的差值是相对较大的，它并没有很好地达成

每名员工在不同天内参与的专班在具体班次上应尽可能平均的要求。所以，相比而言第二种算法的 f_2 直接统筹安排考虑该问题的算式，虽然值较大，但更符合要求。

此外需要说明的是，第一种算法采用枚举算法，其得到的解是精确解，规划所得排班结果即是最优解，排班结果确定。而第二种算法采用随机搜索算法，其得到的解仅为规划的近似解，所对应的排班结果也相对多样，差异较大，处理具体排班时无法确定完全最优的排班结果。从该层面上来说，第一种算法对于具体排班能给出更好的解答。

5 “公平” 条件

5.1 “公平” 的新定义

在本部分，我们按照题干中提到的条件，对“公平”条件的定义进行调整，总排班次数与总工作天数之比尽可能均衡意味着更公平。

类似定义第 i 个人的比值为 T'_i ，表达式如下：

$$T'_i = \frac{A_i + B_i + C_i + D_i + \sum_{j=1}^k (a_{ij} + b_{ij} + c_{ij} + d_{ij})}{W_i} \quad (36)$$

方差为 δ'_1 ，表达式如下：

$$\delta'_1 = \sqrt{\frac{1}{n} \sum_{i=1}^n (T'_i - \frac{\sum_{i=1}^n T'_i}{n})^2} \quad (37)$$

要保持比例一致，此处以初始方差 δ_{10} 为比例参照值，目标函数为 f'_0 ，表达式如下：

$$f'_0 = \frac{\delta'_1}{\delta_{10}} \quad (38)$$

我们利用上一章节探讨的第二种算法，即在多日统筹排班的情况下，考虑在新定义的“公平”条件的情况下，排班情况会发生怎样的变化（具体算法程序见附录，只改变了算法二中的部分条件）。

我们仍然通过随机生成不同数量（1w、10w、100w）的可能排班情况的方法，来测试算法可能达到的效果，同时对结果进行比较和分析来进一步探讨。

在下表中，目标函数的计算公式我们仍然取为 $f = 0.8f'_0 + 0.16f_1 + 0.032f_2 + 0.008f_3$ 。

表 6: 不同量数据下的目标函数值

不同数据量不同次运行结果	f'_0	f_1	f_2	f_3	f
1w 随机数据第 1 次	0.5407	0	1.4269	0.1667	0.4796
10w 随机数据第 1 次	0.4784	0	1.4339	0.0833	0.4293
10w 随机数据第 2 次	0.4425	0	1.2924	0.3333	0.3980
10w 随机数据第 3 次	0.494	0	1.4323	0.0833	0.4417
10w 随机数据第 4 次	0.4938	0	1.4712	0.4167	0.4455
10w 随机数据第 5 次	0.4473	0	1.3746	0.4167	0.4052
100w 随机数据第 1 次	0.4699	0	1.4084	0.1677	0.4223
100w 随机数据第 2 次	0.426	0	1.3728	0.0833	0.3854
100w 随机数据第 3 次	0.4498	0	1.3450	0.2500	0.4049
100w 随机数据第 4 次	0.3881	0	1.3430	0.0833	0.3541
100w 随机数据第 5 次	0.4318	0	1.3254	0.3333	0.3905

同样，根据程序运行的结果，我们发现随着数据量的增大。目标函数会向着更好更小的方向发展。以 100w 随机数据量为参照对象，可以得到此时的目标函数值为 0.4223、0.3854、0.4049、0.3541 和 0.3905。它们的接近程度可以用差距最大值的比例体现： $(0.4223 - 0.3541)/0.4223 = 16.1\%$ 。由于数据选取的随机性，它们的值也应当接近真实的最优解，它们可以被视作该规划问题的近似最优解。则对最优解进行大致估计。限于较大随机样本 100w 情况下已得近似最优解间的差距为 16.1%，而最优值小于近似最优解，则可得：

$$\text{最优解} \geq 0.3541 \times (1 - 16.1\%) = 0.2974 \quad (39)$$

我们选取 1w、10w、100w 数据下各一种的排班情况列在下表：

表 7: 不同量数据下的安排情况

天数	1w 随机数据排班情况	10w 随机数据排班情况	100w 随机数据排班情况
1	A: 1、8 B: 4、10	A: 6、8 B: 2、10	A: 2、6 B: 7、10
2	A: 2、10 B: 5、6	A: 2、5 B: 1、4	A: 4、8 B: 5、10
3	A: 2、10	A: 4、10	A: 2、8
4	A: 5、8 B: 6、11	A: 2、12 B: 8、11	A: 6、10 B: 11、12
5	A: 2、12	A: 11、12	A: 2、12
6	A: 11、14	A: 13、14	A: 11、14
7	A: 13、14	A: 12、15	A: 13、15
8	A: 10、13	A: 14、15	A: 13、14
9	A: 12、15	A: 10、13	A: 14、15

同样，虽然不同数据量下的安排情况不尽相同，但在新公平定义下，从整体来看排班情况依旧维持公平，对于每个人而言其专班次数和各种专班种类也较为均衡。

5.2 公平含义探讨

前文我们列举了两种公平条件以及它们对应的算法和相应运行结果。单从目标函数值来看，新公平定义要优于原有的公平定义，在新公平条件下各位员工可以得到更好的均衡。

但值得注意的是，原有公平定义适用于单日连续排班的枚举算法，该情况下其排班结果可以确定。而新公平定义在单日无法适用，只能考虑连续 k 天的第二种算法，其弊端即在于只能取到近似解，无法唯一确定排班方案。而都考虑第二种算法的近似解，在近似解与真实解的接近程度上，原有定义为 4.6%，而新定义为 16.1%，接近原有的三倍。这说明原有定义下使用该算法能得到较为稳定精确的解，而在新定义下却不尽然。

而有关公平的实现，如我们所知，公平有相对公平和绝对公平。前两种定义均为相对公平，而类似地，我们可以定义绝对公平，如将员工当前累计安排专班次数比当前工作天数的比值与 $\frac{1}{2}$ 的接近程度作为公平条件，类似可更改上述规划和算法得到相应结果，并以此探讨绝对公平与相对公平间的关系。但无论公平条件如何，其在规划中仅代表最首要目标函数的规划，通过规划的求解和算法我们都可以得到相应的（近似）最优解并对该公平条件进行一定评估。

6 新员工分配

按照题设要求，我们还需要探讨新员工分配与专班安排要求间的关系。

依旧采用题设场景，10 名老员工各情况已知，而待分配的有 5 名新员工（序号为 11、12、13、14、15）。此时若不考虑只将新员工全部安置在该车间，而是选择性安排分配。考虑专班安排要求，我们希望新员工安置后车间的可排班天数尽可能多，且目标函数值尽可能小。但由前分析讨论，可排班天数越多，相应的目标函数值 f 会增大，所以此处引入比值 $f : \text{可排班天数 } n$ 来体现不同新员工方案后的影响。同时，此处考虑采用第二种多天统筹安排算法，取 100w 随机数据，以一次的运行结果来粗略表示新员工引入后的 f 值。由于某几种新员工方案差别不大，我们仅对其中几种代表情况代入运行，得到如下结果：

表 8: 不同新员工方案下的结果

新员工编号	可排班天数 n	目标函数 f	$f : n$
11	5	0.3841	0.07682
11,12	5	0.4356	0.08712
11,12,13,14	8	0.4937	0.06171
11,12,13,15	7	0.4685	0.06693
12,13,14,15	8	0.4382	0.05478
11,12,13,14,15	9	0.5021	0.05579

上述表格即给出了题设场景下选择不同新员工方案对于排班要求的影响，其影响用 $f, n, f : n$ 三个参量来衡量。我们可以发现在新员工数量较多时，其可排班天数也相对较多。所以若以可排班天数为专班安排指标，会在不超出上限情况下尽量引入多的员工。而若以排班公平、均衡为考虑参数，则衡量指标为 $f : n$ ，该场景下的最优新员工方案为选择编号 12、13、14、15 的员工。其余场景可采用类似方法，对新员工方案分类并代入算法求值进行分析，最终得出最优的那个新员工分配方案。

7 算法分析与优化

7.1 可行性分析

这部分，我们首先要说明我们取定目标函数中权重系数 $\lambda_0 = 0.8, \lambda_1 = 0.16, \lambda_2 = 0.032, \lambda_3 = 0.008$ 的合理性。由于权重系数主要在第一种算法中就已确定，我们将其放在第一种算法情境下考虑。通过比较各子目标函数该比例系数下值与其单独最优值的差别我们来判断说明该权重系数组合是否可行。由于 f_1 和 f_3 本身下界为 0，表 3 中所得 f_1 和 f_3 值大部分均已取得 0，则认为它们是接近的，权重系数对于 f_1 和 f_3 来说可行。

考虑 f_0 ，我们要计算其第一种算法下的最优值。改变待定系数使得目标函数的计算公式为 $f = f_0$ ，所得结果如表 9。

表 9: 各天专班安排情况 ($f = f_0$)

天数	排班情况	f_0	f
1	A: 6、10 B: 2、3	0.3068	0.3068
2	A: 6、8 B: 2、5	0.2939	0.2939
3	A: 5、6	0.2873	0.2873
4	A: 10、11 B: 4、8	0.2568	0.2568
5	A: 2、12	0.2068	0.2068
6	A: 10、11	0.2099	0.2099
7	A: 13、14	0.1276	0.1276
8	A: 11、12	0.1196	0.1196
9	A: 11、12	0.1506	0.1506

对比表 3 与表 9 数据，其 f_0 值总体相当接近，最大差值在第 8 天，但此时表 3 数据优于表 9，说明所取比例系数合理。

考虑 f_2 ，同样要计算其第一种算法下的最优值。改变待定系数使得目标函数的计算公式为 $f = f_2$ ，所得结果如下表：

表 10: 各天专班安排情况 ($f = f_2$)

天数	排班情况	f_2	f
1	A: 6、9 B: 1、8	1	1
2	A: 4、10 B: 6、9	1.1395	1.1395
3	A: 8、11	1.0575	1.0575
4	A: 6、12 B: 10、11	1.1096	1.1096
5	A: 2、11	1.2029	1.2029
6	A: 13、14	0.9191	0.9191
7	A: 10、13	1.1798	1.1798
8	A: 11、14	1.2095	1.2095
9	A: 11、14	1.2677	1.2677

对比表 3 与表 10 数据，其 f_2 值总体相当接近，最大差值在第 1 天，相差比例为 $(1.2791 - 1)/1 = 27.91\% < 50\%$ 。当 f_2 作为第三目标函数值时，其影响不大，权重系数合理。所以综上讨论，我们补充说明了我们的目标函数公式中权重取值的合理性。

此外，我们要说明第二种随机搜索算法在解决该规划问题上的合理性。我们知道在运用随机搜索等启发式算法时求得的只能是规划的近似解，其建立的基础是大量的数据。在具体随机搜索中，为确保能有大量数据，我们要保证能通过规划原有约束条件的随机数据不能太少。考虑该问题，我们发现它的复杂之处主要在于目标，而非约束。其约束仅在于驾驶资格。而根据所给场景，其中具有驾驶资格的员工不在少数，所以随机数据能通过约束的肯定也不在少数。而事实上，在取 100w 组随机数据时，根据计算机的运行显示，其中能通过约束的有 14w 余种，比例达 $\frac{1}{7}$ ，该比例基本稳定且已能保证大数据要求。

7.2 现存问题

在前文中我们主要分析了算法的优势与可行性，但我们的算法本身也有许多问题与局限，这主要表现在以下方面：

1、我们的算法都是基于题中所给的特定条件，各权重系数和算法仅在该场景下说明可行，尚未清楚对于其余场景是否可行，算法适用范围较窄；

2、该场景下无 C、D 班次，我们规划和算法中所设想的有关 C、D 排班的部分均未涉及，无法体现算法的全面性；

3、在多天统筹安排时，如果利用穷举法其原有计算复杂度是呈指数级别增长的，在 $k = 9$ 时，其数量级非常大。随机搜索算法中百万级别的随机数据虽有一定可行性，但所占比例仍过小，不够代表全体情况，所得近似解与最优解间的逼近性未知。如第二种公平条件应用时就出现了较大差距。因而近似解是否能代表规划最优解有时存疑。

7.3 改进策略

针对以上算法的现存问题，我们也有初步的改进策略：

1、在该问题背景下考虑更多的具体场景并将场景代入算法，训练算法从而得到更为优越的权重取值。同时可取含 C、D 排班的具体场景，从而体现我们所取规划和算法的优越性；

2、考虑应用更优化的启发式算法或近似算法来处理多天统筹问题中的指数级情况，如遗传算法、蚁群算法等。

8 结语

在本题的探究过程中，我们建立了数学规划的模型，并针对不同要求设计了两种算法，总体而言可以较好地满足题目的要求，解决了相应的问题，并且有一定的实际应用价值。对于探究过程中出现的问题，我们也进行了分析并提出了相应的解决策略，但因为时间关系没有进行更深入的探讨，恳请批评指正！

附录

算法一

主函数

```
%初始条件，后续会有w{i}，表示过去了i-1天的员工基本量情况
x=1;
%规划的天数
n=zeros(1,x);
%n为x天内各天在岗员工人数变化
N=zeros(1,x);
%N为每天班次
for i=1:(x+1)
    w{i+1}=w{i};
    for j=1:15
        if(w{i}(j,1)>0&&w{i}(j,2)>w{i}(j,3))
            w{i+1}(j,3)=w{i+1}(j,3)+1;
        end
    end
end
%各天的工作人员已工作天数变化赋值
for i=1:x
    for j=1:15
        if(w{i}(j,1)>0&&w{i}(j,3)>=0&&w{i}(j,2)>w{i}(j,3))
            n(1,i)=n(1,i)+1;
        end
    end
    N(1,i)=transnumber(n(1,i));
end
%规划天内每日工作人员变化与班数变化
for i=1:x
    arr{i}=zeros(15,9);
end
%排班矩阵初始化（后四列为排班有效值，其他都写0，便于矩阵加减）

%主函数
%A
if(N(1)==1)
    f=1000000000000;
    i1=1;
    while(i1<=14)
        if(w{1}(i1,3)<0||w{1}(i1,3)==w{1}(i1,2))
            i1=i1+1;
            continue;
        end
        i2=i1+1;
        while(i2<=15)
            if(w{1}(i2,3)<0||w{1}(i2,3)==w{1}(i2,2))
                i2=i2+1;
                continue;
            end
            if(w{1}(i2,4)==0&&w{1}(i1,4)==0)
                i2=i2+1;
                continue;
            end
        end
    end
end
```

```

        end
        for i=1:x
            arr{i}=zeros(15,9);
        end
        arrt=arr;
        arr{1}(i1,6)=1;
        arr{1}(i2,6)=1;
        w{3}=w{2};
        w{2}=w{2}+arr{1};
        M=F(1,w)*1+(1-f1(1,w))*0+f2(1,w)*0+(1-f3(1,arr,w))*0;
        w{2}=w{3};
        if(M>f)
            arr=arrt;
        else
            f=M;
            brr=arr;
        end
        i2=i2+1;
    end
    i1=i1+1;
end
end

%AB
if(N(1)==2)
    f=1000000000000;
    i1=1;
    while(i1<=14)
        if(w{1}(i1,3)<0 || w{1}(i1,3)==w{1}(i1,2))
            i1=i1+1;
            continue;
        end
        i2=i1+1;
        while(i2<=15)
            if(w{1}(i2,3)<0 || w{1}(i2,3)==w{1}(i2,2))
                i2=i2+1;
                continue;
            end
            if(w{1}(i2,4)==0&&w{1}(i1,4)==0)
                i2=i2+1;
                continue;
            end
            j1=1;
            while(j1<=14)
                if(w{1}(j1,3)<0 || w{1}(j1,3)==w{1}(j1,2) || j1==i1 || j1==i2)
                    j1=j1+1;
                    continue;
                end
                j2=j1+1;
                while(j2<=15)
                    for i=1:x
                        arr{i}=zeros(15,9);
                    end
                    arrt=arr;
                    if(w{1}(j2,3)<0 || w{1}(j2,3)==w{1}(j2,2) || j2==i1 || j2==i2)
                        j2=j2+1;
                        continue;
                    end
                end
            end
        end
    end
end

```

```

        if(w{1}(j2,4)==0&&w{1}(j1,4)==0)
            j2=j2+1;
            continue;
        end
        arr{1}(i1,6)=1;
        arr{1}(i2,6)=1;
        arr{1}(j1,7)=1;
        arr{1}(j2,7)=1;
        w{3}=w{2};
        w{2}=w{2}+arr{1};
        M=F(1,w)*0.8+(1-f1(1,w))*0.16+f2(1,w)*0.032+(1-
f3(1,arr,w))*0.008;
        w{2}=w{3};
        if(M>f)
            arr=arrt;
        else
            f=M;
            brr=arr;
        end
        j2=j2+1;
    end
    j1=j1+1;
end
i2=i2+1;
end
i1=i1+1;
end
end

%ABC
if(N(1)==3)
    f=1000000000000;
    i1=1;
    while(i1<=14)
        if(w{1}(i1,3)<0 || w{1}(i1,3)==w{1}(i1,2))
            i1=i1+1;
            continue;
        end
        i2=i1+1;
        while(i2<=15)
            if(w{1}(i2,3)<0 || w{1}(i2,3)==w{1}(i2,2))
                i2=i2+1;
                continue;
            end
            if(w{1}(i2,4)==0&&w{1}(i1,4)==0)
                i2=i2+1;
                continue;
            end
            j1=1;
            while(j1<=14)
                if(w{1}(j1,3)<0 || w{1}(j1,3)==w{1}(j1,2) || j1==i1 || j1==i2)
                    j1=j1+1;
                    continue;
                end
                j2=j1+1;
                while(j2<=15)
                    for i=1:x
                        arr{i}=zeros(15,9);

```

```

end
arrt=arr;
if(w{1}(j2,3)<0 || w{1}(j2,3)==w{1}(j2,2) || j2==i1 || j2==i2)
    j2=j2+1;
    continue;
end
if(w{1}(j2,4)*w{1}(j1,4)==0)
    j2=j2+1;
    continue;
end
k1=1;
while(k1<=14)
    if(w{1}(k1,3)<0 || w{1}(k1,3)==w{1}(
(k1,2) || k1==i1 || k1==i2 || k1==j1 || k1==j2)
        k1=k1+1;
        continue;
    end
    k2=k1+1;
    while(k2<=15)
        for i=1:x
            arr{i}=zeros(15,9);
        end
        arrt=arr;
        if(w{1}(k2,3)<0 || w{1}(k2,3)==w{1}(
(k2,2) || k2==i1 || k2==i2 || k2==j1 || k2==j2 || k2==k1)
            k2=k2+1;
            continue;
        end
        if(w{1}(k2,4)==0&&w{1}(k1,4)==0)
            k2=k2+1;
            continue;
        end
        for i=1:x
            arr{i}=zeros(15,9);
        end
        arrt=arr;
        arr{1}(i1,6)=1;
        arr{1}(i2,6)=1;
        arr{1}(j1,7)=1;
        arr{1}(j2,7)=1;
        arr{1}(k1,7)=1;
        arr{1}(k2,7)=1;
        w{3}=w{2};
    w{2}=w{2}+arr{1};
    M=F(1,w)*1+(1-f1(1,w))*0+f2(1,w)*0+(1-f3(1,arr,w))*0;
    w{2}=w{3};
    if(M>f)
        arr=arrt;
    else
        f=M;
        brr=arr;
    end
    k2=k2+1;
end
k1=k1+1;
end
j2=j2+1;
end
end

```

```

        j1=j1+1;
    end
    i2=i2+1;
end
i1=i1+1;
end
end

%ABCD
if(N(1)==4)
    f=1000000000000;
    i1=1;
    while(i1<=14)
        if(w{1}(i1,3)<0 || w{1}(i1,3)==w{1}(i1,2))
            i1=i1+1;
            continue;
        end
        i2=i1+1;
        while(i2<=15)
            if(w{1}(i2,3)<0 || w{1}(i2,3)==w{1}(i2,2))
                i2=i2+1;
                continue;
            end
            if(w{1}(i2,4)==0&&w{1}(i1,4)==0)
                i2=i2+1;
                continue;
            end
            j1=1;
            while(j1<=14)
                if(w{1}(j1,3)<0 || w{1}(j1,3)==w{1}(j1,2) || j1==i1 || j1==i2)
                    j1=j1+1;
                    continue;
                end
                j2=j1+1;
                while(j2<=15)
                    for i=1:x
                        arr{i}=zeros(15,9);
                    end
                    arrt=arr;
                    if(w{1}(j2,3)<0 || w{1}(j2,3)==w{1}(j2,2) || j2==i1 || j2==i2)
                        j2=j2+1;
                        continue;
                    end
                    if(w{1}(j2,4)==0&&w{1}(j1,4)==0)
                        j2=j2+1;
                        continue;
                    end
                    k1=1;
                    while(k1<=14)
                        if(w{1}(k1,3)<0 || w{1}(k1,3)==w{1}(
(k1,2) || k1==i1 || k1==i2 || k1==j1 || k1==j2)
                            k1=k1+1;
                            continue;
                        end
                        k2=k1+1;
                        while(k2<=15)
                            for i=1:x
                                arr{i}=zeros(15,9);

```

```

end
arrr=arr;
if(w{1}(k2,3)<0||w{1}(k2,3)==w{1}
(k2,2)||k2==i1||k2==i2||k2==j1||k2==j2||k2==k1)
    k2=k2+1;
    continue;
end
if(w{1}(k2,4)==0&&w{1}(k1,4)==0)
    k2=k2+1;
    continue;
end
for i=1:x
    arr{i}=zeros(15,9);
end
l1=1;
while(l1<=14)
    if(w{1}(l1,3)<0||w{1}(l1,3)==w{1}
(l1,2)||l1==i1||l1==i2||l1==j1||l1==j2||l1==k1||l1==k2)
        l1=l1+1;
        continue;
    end
    l2=l1+1;
    while(l2<=15)
        for i=1:x
            arr{i}=zeros(15,9);
        end
        arrr=arr;
        if(w{1}(l2,3)<0||w{1}(l2,3)==w{1}
(l2,2)||l2==i1||l2==i2||l2==j1||l2==j2||l2==k1||l2==k2)
            l2=l2+1;
            continue;
        end
        if(w{1}(l2,4)==0&&w{1}(l1,4)==0)
            l2=l2+1;
            continue;
        end
        for i=1:x
            arr{i}=zeros(15,9);
        end
        arrr=arr;
        arr{1}(i1,6)=1;
        arr{1}(i2,6)=1;
        arr{1}(j1,7)=1;
        arr{1}(j2,7)=1;
        arr{1}(k1,7)=1;
        arr{1}(k2,7)=1;
        arr{1}(l1,7)=1;
        arr{1}(l2,7)=1;
        w{3}=w{2};
        w{2}=w{2}+arr{1};
        M=F(1,w)*0.8+(1-f1(1,w))*0.16+f2(1,w)*0.032+
(1-f3(1,arr,w))*0.008;

        w{2}=w{3};
        if(M>f)
            arr=arrr;
        else
            f=M;
            brr=arr;

```

```

                                end
                                l2=l2+1;
                            end
                            l1=l1+1;
                        end
                        k2=k2+1;
                    end
                    k1=k1+1;
                end
                j2=j2+1;
            end
            j1=j1+1;
        end
        i2=i2+1;
    end
    i1=i1+1;
end

w{2}=w{2}+brr{1};
a=f1(1,w);
b=f2(1,w);
c=f3(1,brr,w);
d=F(1,w);

```

子函数

```

%下列为用到函数
function y=bool(x)
    if x>0
        y=1;
    else
        y=0;
    end
end
%判断有无函数bool

```

```

function y=transnumber(x)
    if(x>=5&&x<=7)
        y=1;
    elseif(x>=8&&x<=10)
        y=2;
    elseif(x>=11&&x<=12)
        y=3;
    else
        y=4;
    end
end
%人数求出排班数

```

```

function y=f1(x,w)
    n=zeros(1,x);
    %n为x天内各天在岗员工人数变化
    N=zeros(1,x);
    %N为每天班次
    for i=1:x

```

```

        for j=1:15
            if(w{i}(j,1)>0&&w{i}(j,3)>=0&&w{i}(j,2)>w{i}(j,3))
                n(1,i)=n(1,i)+1;
            end
        end
        N(1,i)=transnumber(n(1,i));
    end
    NA=0;
    for i=1:x
        NA=NA+N(1,i);
        %总班数
    end
    NB=NA;
    %符合已经有专班经验员工的班数
    for k=1:x
        for i=1:14
            for j=(i+1):15
                NB=NB-(w{k+1}(i,6)-w{k}(i,6))*(1-bool(w{k}(i,6)))*(w{k+1}(j,6)-w{k}(j,6))*(1-bool(w{k}(i,6)))-(w{k+1}(i,7)-w{k}(i,7))*(1-bool(w{k}(i,7)))*(w{k+1}(j,7)-w{k}(j,7))*(1-bool(w{k}(i,7)))-(w{k+1}(i,8)-w{k}(i,8))*(1-bool(w{k}(i,8)))*(w{k+1}(j,8)-w{k}(j,8))*(1-bool(w{k}(i,8)))-(w{k+1}(i,9)-w{k}(i,9))*(1-bool(w{k}(i,9)))*(w{k+1}(j,9)-w{k}(j,9))*(1-bool(w{k}(i,9)));
            end
        end
    end
    y=NB/NA;
end
%条件1目标函数

```

```

function y=f2(x,w)
    n=zeros(1,x);
    %n为x天内各天在岗员工人数变化
    N=zeros(1,x);
    %N为每天班次
    for i=1:x
        for j=1:15
            if(w{i}(j,1)>0&&w{i}(j,3)>=0&&w{i}(j,2)>w{i}(j,3))
                n(1,i)=n(1,i)+1;
            end
        end
        N(1,i)=transnumber(n(1,i));
    end
    n0=0;
    %最后一天参加工作至少一天人数的初始化
    n1=0;
    %第一天参加工作
    var0=0;
    var1=0;
    %方差均值初始化
    for i=1:15
        if(w{x+1}(i,3)>0)
            n0=n0+1;
        end
        if(w{1}(i,3)>0)
            n1=n1+1;
        end
    end
end

```



```

for i=1:15
    var0=var0+var([w{x+1}(i,6), w{x+1}(i,7), w{x+1}(i,8), w{x+1}(i,9)]);
end
var0=var0/n0;
for i=1:15
    var1=var1+var([w{1}(i,6), w{1}(i,7), w{1}(i,8), w{1}(i,9)]);
end
var1=var1/n1;
y=var0/var1;
end
%条件2 目标函数

```

```

function y=f3dis(i,j,x,arr,w)
    if((arr{x}(i,6)*arr{x}(j,6)>0 || arr{x}(i,7)*arr{x}(j,7)>0 || arr{x}(i,8)*arr{x}(j,8)>0 || arr{x}(i,9)*arr{x}(j,9)>0)&&(w{1}(i,5)<=1 || w{1}(j,5)<=1 || (w{1}(i,5)>=2&&w{1}(i,5)<=3&&w{1}(j,5)>=2&&w{1}(j,5)<=3)))
        y=1;
    else
        y=0;
    end
end
%条件3成立判断函数

```

```

function y=f3(x,arr,w)
    n=zeros(1,x);
    %n为x天内各天在岗员工人数变化
    N=zeros(1,x);
    %N为每天班次
    for i=1:x
        for j=1:15
            if(w{i}(j,1)>0&&w{i}(j,3)>=0&&w{i}(j,2)>w{i}(j,3))
                n(1,i)=n(1,i)+1;
            end
        end
        N(1,i)=transnumber(n(1,i));
    end
    NA=0;
    for i=1:x
        NA=NA+N(1,i);
        %总班数
    end
    NB=0;
    %符合的班数
    for k=1:x
        for i=1:14
            for j=(i+1):15
                NB=NB+f3dis(i,j,x,arr,w);
            end
        end
    end
    y=NB/NA;
end
%f3目标函数

```

```

function y=F(x,w)

```

```

n=zeros(1,x);
%n为x天内各天在岗员工人数变化
N=zeros(1,x);
%N为每天班次
for i=1:x
    for j=1:15
        if(w{i}(j,1)>0&&w{i}(j,3)>=0&&w{i}(j,2)>w{i}(j,3))
            n(1,i)=n(1,i)+1;
        end
    end
    N(1,i)=transnumber(n(1,i));
end
y=0;
ratio=zeros(15,x);
%比例数存放矩阵
for i=1:15
    for j=1:x
        if(w{j+1}(i,3)>0)
            ratio(i,j)=(w{j+1}(i,6)+w{j+1}(i,7)+w{j+1}(i,8)+w{j+1}(i,9))/w{j+1}(i,3);
        else
            ratio(i,j)=0;
        end
    end
end
Q=zeros(1,x);
%定义每天ratio的方差
for i=1:x
    a=0;
    b=0;
    for j=1:15
        if(ratio(j,i)>0)
            a=a+1;
            b=b+ratio(j,i);
        end
    end
    for j=1:15
        Q(i)=Q(i)+(ratio(j,i)-b/a)^2/a;\
    end
end
for i=1:x
    y=y+Q(i);
end
y=(y/x)^0.5;
end
%最重要先决条件

```

算法二

主函数

```

%主问题的多天排班规划
k=9;n=15;
%k=input('总天数: ');
%n=input('总人数: ');
n0=zeros(1,k);%第j天员工数量
m0=zeros(1,k);%到第j天已工作过员工数量

```

```

%W=input('工作天数: ');%n*1
%w=input('已工作天数: ');%n*1
%q=input('是否有驾驶资格: ');%n*1
%t=input('历史事故次数: ');%n*1
%A=input('已排A班次: ');%n*1
%B=input('已排B班次: ');
%C=input('已排C班次: ');
%D=input('已排D班次: '); %部分注释内容修改后可适用于任何场景
w=[6;11;9;10;13;15;6;10;5;10;6;10;4;8;5];
w=[4;6;8;5;8;11;4;5;3;1;-2;-3;-5;-5;-6];
q=[1;1;1;1;1;1;1;0;0;0;1;0;1;0;1];
t=[0;0;0;1;1;2;1;2;1;0;0;0;0;0;0];
A=[2;1;1;1;2;1;1;1;0;0;0;0;0;0;0];
B=[0;1;2;1;1;1;1;0;0;0;0;0;0;0;0];
C=[0;0;1;1;0;1;0;1;1;0;0;0;0;0;0];
D=[0;0;0;0;1;1;0;0;1;0;0;0;0;0;0];
A1=zeros(n,k);B1=zeros(n,k);C1=zeros(n,k);D1=zeros(n,k);
x=zeros(n,k);
y=zeros(n,k);
z=zeros(n,k);
a1=zeros(1,k); %第j天总共要的人数
b1=zeros(1,k); c1=zeros(1,k); d1=zeros(1,k);
a=zeros(n,k);b=zeros(n,k);c=zeros(n,k);d=zeros(n,k);
for i=1:n
    for j=1:k
        if 1-w(i,1)<=j &&j<=w(i,1)-w(i,1)
            x(i,j)=1;
        end
    end
end
for j=1:k
    for i=1:n
        n0(1,j)=n0(1,j)+x(i,j);
    end
end
M=
[0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0;2,0,0,0;2,0,0,0;2,2,0,0;2,2,0,0;2,2,0,0
;2,2,2,0;2,2,2,0;2,2,2,2;2,2,2,2;2,2,2,2];
for j=1:k
    if n0(1,j)<=15
        a1(j)=M(n0(1,j),1);
        b1(j)=M(n0(1,j),2);
        c1(j)=M(n0(1,j),3);
        d1(j)=M(n0(1,j),4);
    end
end
for i=1:n
    for j=1:k
        z(i,j)=iszero(w(i,1)+1)*w(i,1);
        for l=1:j
            z(i,j)=z(i,j)+x(i,l);
        end
    end
end
for j=1:k
    for i=1:n
        m0(1,j)=m0(1,j)+iszero(z(i,j));
    end
end

```

```

end %数据预处理
fmin=100000;
f0min=100000;
f1min=100000;
f2min=100000;
f3min=100000;
amin=a;bmin=b;cmin=c;dmin=d;
for cic=1:1000000 %搜索次数, 10000,100000,1000000
    ss=1;
    a=zeros(n,k);b=zeros(n,k);c=zeros(n,k);d=zeros(n,k);
    for j=1:k
        x1=zeros(n0(j),1);
        cnt=1;
        for i=1:n
            if x(i,j)>0
                x1(cnt,1)=i;
                cnt=cnt+1;
            end
        end
        yy(j)=a1(j)+b1(j)+c1(j)+d1(j);
        ri=randperm(n0(j),yy(j));
        if a1(j)==2
            a(x1(ri(1),1),j)=1;
            a(x1(ri(2),1),j)=1;
        end
        if b1(j)==2
            b(x1(ri(3),1),j)=1;
            b(x1(ri(4),1),j)=1;
        end
        if c1(j)==2
            c(x1(ri(5),1),j)=1;
            c(x1(ri(6),1),j)=1;
        end
        if d1(j)==2
            d(x1(ri(7),1),j)=1;
            d(x1(ri(8),1),j)=1;
        end
        %在满足基本排班需求下的随机量取值
        if dot(a(:,j),q)<a1(j)/2 || dot(b(:,j),q)<b1(j)/2 || dot(c(:,j),q)
<c1(j)/2 || dot(d(:,j),q)<d1(j)/2
            ss=0;
        end %约束条件
    end
    if ss==1
        result=f(n,k,A,B,C,D,a,b,c,d,z,m0,a1,b1,c1,d1,t,w);
        if result<fmin
            fmin=result;
            f0min=f0(n,k,A,B,C,D,a,b,c,d,z,m0,w);
            f1min=f1(n,k,A,B,C,D,a,b,c,d,a1,b1,c1,d1);
            f2min=f2(n,k,a,b,c,d,A,B,C,D);
            f3min=f3(n,k,a,b,c,d,t,a1,b1,c1,d1);
            amin=a;
            bmin=b;
            cmin=c;
            dmin=d;
        end
    end %判断目标函数最小值
end
disp(fmin);

```

```

disp(f0min);
disp(f1min);
disp(f2min);
disp(f3min);
disp(amin);
disp(bmin);
disp(cmin);
disp(dmin);

```

改变公平条件后:

```

%第二种公平条件下的排班规划算法
k=9;n=15;
%k=input('总天数: ');
%n=input('总人数: ');
n0=zeros(1,k);%第j天员工数量
m0=zeros(1,k);%到第j天已工作过员工数量
%w=input('工作天数: ');%n*1
%w=input('已工作天数: ');%n*1
%q=input('是否有驾驶资格: ');%n*1
%t=input('历史事故次数: ');%n*1
%A=input('已排A班次: ');%n*1
%B=input('已排B班次: ');
%C=input('已排C班次: ');
%D=input('已排D班次: '); %部分注释内容修改后可适用于任何场景
w=[6;11;9;10;13;15;6;10;5;10;6;10;4;8;5];
w=[4;6;8;5;8;11;4;5;3;1;-2;-3;-5;-5;-6];
q=[1;1;1;1;1;1;1;0;0;0;1;0;1;0;1];
t=[0;0;0;1;1;2;1;2;1;0;0;0;0;0;0];
A=[2;1;1;1;2;1;1;1;0;0;0;0;0;0;0];
B=[0;1;2;1;1;1;1;0;0;0;0;0;0;0;0];
C=[0;0;1;1;0;1;0;1;1;0;0;0;0;0;0];
D=[0;0;0;0;1;1;0;0;1;0;0;0;0;0;0];
A1=zeros(n,k);B1=zeros(n,k);C1=zeros(n,k);D1=zeros(n,k);
x=zeros(n,k);
y=zeros(n,k);
z=zeros(n,k);
a1=zeros(1,k); %第j天总共要的人数
b1=zeros(1,k); c1=zeros(1,k); d1=zeros(1,k);
a=zeros(n,k);b=zeros(n,k);c=zeros(n,k);d=zeros(n,k);
for i=1:n
    for j=1:k
        if 1-w(i,1)<=j &&j<=w(i,1)-w(i,1)
            x(i,j)=1;
        end
    end
end
for j=1:k
    for i=1:n
        n0(1,j)=n0(1,j)+x(i,j);
    end
end
M=
[0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0;2,0,0,0;2,0,0,0;2,0,0,0;2,2,0,0;2,2,0,0;2,2,0,0;
2,2,2,0;2,2,2,0;2,2,2,2;2,2,2,2;2,2,2,2];
for j=1:k
    if n0(1,j)<=15

```

```

        a1(j)=M(n0(1,j),1);
        b1(j)=M(n0(1,j),2);
        c1(j)=M(n0(1,j),3);
        d1(j)=M(n0(1,j),4);
    end
end
for i=1:n
    for j=1:k
        z(i,j)=isZero(w(i,1)+1)*w(i,1);
        for l=1:j
            z(i,j)=z(i,j)+x(i,l);
        end
    end
end
for j=1:k
    for i=1:n
        m0(1,j)=m0(1,j)+isZero(z(i,j));
    end
end
end %数据预处理
fmin=100000;
ff0min=100000;
f1min=100000;
f2min=100000;
f3min=100000;
amin=a;bmin=b;cmin=c;dmin=d;
for cic=1:1000000 %搜索次数, 10000,100000,1000000
    ss=1;
    a=zeros(n,k);b=zeros(n,k);c=zeros(n,k);d=zeros(n,k);
    for j=1:k
        x1=zeros(n0(j),1);
        cnt=1;
        for i=1:n
            if x(i,j)>0
                x1(cnt,1)=i;
                cnt=cnt+1;
            end
        end
        yy(j)=a1(j)+b1(j)+c1(j)+d1(j);
        ri=randperm(n0(j),yy(j));
        if a1(j)==2
            a(x1(ri(1),1),j)=1;
            a(x1(ri(2),1),j)=1;
        end
        if b1(j)==2
            b(x1(ri(3),1),j)=1;
            b(x1(ri(4),1),j)=1;
        end
        if c1(j)==2
            c(x1(ri(5),1),j)=1;
            c(x1(ri(6),1),j)=1;
        end
        if d1(j)==2
            d(x1(ri(7),1),j)=1;
            d(x1(ri(8),1),j)=1;
        end %在满足基本排班需求下的随机量取值
        if dot(a(:,j),q)<a1(j)/2 || dot(b(:,j),q)<b1(j)/2 || dot(c(:,j),q)
<c1(j)/2 || dot(d(:,j),q)<d1(j)/2
            ss=0;

```

```

        end %约束条件
    end
    if ss==1
        result=g(n,k,A,B,C,D,a,b,c,d,z,m0,a1,b1,c1,d1,t,w,w);
        if result<fmin
            fmin=result;
            ff0min=ff0(n,k,A,B,C,D,a,b,c,d,z,m0,w,w);
            f1min=f1(n,k,A,B,C,D,a,b,c,d,a1,b1,c1,d1);
            f2min=f2(n,k,a,b,c,d,A,B,C,D);
            f3min=f3(n,k,a,b,c,d,t,a1,b1,c1,d1);
            amin=a;
            bmin=b;
            cmin=c;
            dmin=d;
        end
    end %判断目标函数最小值
end
disp(fmin);
disp(ff0min);
disp(f1min);
disp(f2min);
disp(f3min);
disp(amin);
disp(bmin);
disp(cmin);
disp(dmin);

```

子函数

```

function y = isZero(x)
    if x>0
        y=1;
    else
        y=0;
    end
end

```

```

function y = f0(n,k,A,B,C,D,a,b,c,d,z,m0,w)
    T=zeros(n,k); %比值
    dt=zeros(1,k); %方差
    ww=zeros(1,k); %权重
    for j=1:k
        for i=1:n
            T(i,j)=A(i,1)+B(i,1)+C(i,1)+D(i,1);
            for l=1:j
                T(i,j)=T(i,j)+a(i,l)+b(i,l)+c(i,l)+d(i,l);
            end
            T(i,j)=T(i,j)/(z(i,j)+0.00001);
        end
        average=0;
        for i=1:n
            average=average+T(i,j);
        end
        average=average/m0(1,j);
        for i=1:n
            dt(1,j)=dt(1,j)+isZero(z(i,j))*(T(i,j)-average)^2;
        end
    end
end

```

```

        dt(1,j)=sqrt(dt(1,j)/m0(1,j));
    end
    for j=1:k
        ww(1,j)=(2*k+1-2*j)*1.0/(k^2);
    end
    T0=zeros(n,1);mm0=0;
    for i=1:n
        T0(i,1)=(A(i,1)+B(i,1)+C(i,1)+D(i,1))/(w(i,1)+0.00001);
        mm0=mm0+isZero(w(i,1));
    end
    dt0=0;
    average0=0;
    for i=1:n
        average0=average0+T0(i,1);
    end
    average0=average0/mm0;
    for i=1:n
        dt0=dt0+isZero(w(i,1))*(T0(i,1)-average0)^2;
    end
    dt0=sqrt(dt0/mm0);
    fenzi=dot(dt,ww);
    y=fenzi*1.0/dt0;
end

```

```

function y = ff0(n,k,A,B,C,D,a,b,c,d,z,m0,w,w)
    T=zeros(n,1);
    for i=1:n
        T(i,1)=A(i,1)+B(i,1)+C(i,1)+D(i,1);
        for j=1:k
            T(i,1)=T(i,1)+a(i,j)+b(i,j)+c(i,j)+d(i,j);
        end
        T(i,1)=T(i,1)*1.0/w(i,1);
    end
    average=0;
    for i=1:n
        average=average+T(i,1);
    end
    average=average/n;
    dt1=0;
    for i=1:n
        dt1=dt1+(T(i,1)-average)^2;
    end
    dt1=sqrt(dt1/n);
    T0=zeros(n,1);mm0=0;
    for i=1:n
        T0(i,1)=(A(i,1)+B(i,1)+C(i,1)+D(i,1))/(w(i,1)+0.00001);
        mm0=mm0+isZero(w(i,1));
    end
    dt0=0;
    average0=0;
    for i=1:n
        average0=average0+T0(i,1);
    end
    average0=average0/mm0;
    for i=1:n
        dt0=dt0+isZero(w(i,1))*(T0(i,1)-average0)^2;
    end
end

```



```

dt0=sqrt(dt0/mm0);
y=dt1/dt0;
end

```

```

function y = f1(n,k,A,B,C,D,a,b,c,d,a1,b1,c1,d1)
A1=zeros(n,k);B1=zeros(n,k);C1=zeros(n,k);D1=zeros(n,k);
for j=1:k
    for i=1:n
        A1(i,j)=A(i,1);B1(i,j)=B(i,1);C1(i,j)=C(i,1);D1(i,j)=D(i,1);
        for l=1:j
            A1(i,j)=A1(i,j)+a(i,l);B1(i,j)=B1(i,j)+b(i,l);C1(i,j)=C1(i,j)+c(i,l);D1(i,j)=D1(i,j)+d(i,l);
        end

        A1(i,j)=isZero(A1(i,j));B1(i,j)=isZero(B1(i,j));C1(i,j)=isZero(C1(i,j));D1(i,j)=isZero(D1(i,j));
    end
end
U=0;
for j=1:k
    for i1=1:n-1
        for i2=i1+1:n
            U=U+a(i1,j)*a(i2,j)*(1-A1(i1,j))*(1-A1(i2,j))+b(i1,j)*b(i2,j)*(1-B1(i1,j))*(1-B1(i2,j))+c(i1,j)*c(i2,j)*(1-C1(i1,j))*(1-C1(i2,j))+d(i1,j)*d(i2,j)*(1-D1(i1,j))*(1-D1(i2,j));
        end
    end
end
fenmu=0;
for j=1:k
    fenmu=fenmu+(a1(1,j)+b1(1,j)+c1(1,j)+d1(1,j))/2;
end
y=U*1.0/fenmu;
end

```

```

function y = f2(n,k,a,b,c,d,A,B,C,D)
N=zeros(n,1);
aa=zeros(n,1);bb=zeros(n,1);cc=zeros(n,1);dd=zeros(n,1);
dt2=zeros(n,1);
for i=1:n
    N(i,1)=A(i,1)+B(i,1)+C(i,1)+D(i,1);
    aa(i,1)=A(i,1);bb(i,1)=B(i,1);cc(i,1)=C(i,1);dd(i,1)=D(i,1);
    for j=1:k
        N(i,1)=N(i,1)+a(i,j)+b(i,j)+c(i,j)+d(i,j);
        aa(i,1)=aa(i,1)+a(i,j);bb(i,1)=bb(i,1)+b(i,j);
        cc(i,1)=cc(i,1)+c(i,j);dd(i,1)=dd(i,1)+d(i,j);
    end
    N(i,1)=N(i,1)*1.0/4;
    dt2(i,1)=sqrt(0.25*((aa(i,1)-N(i,1))^2+(bb(i,1)-N(i,1))^2+(cc(i,1)-N(i,1))^2+(dd(i,1)-N(i,1))^2));
end
N0=zeros(n,1);dt20=zeros(n,1);
for i=1:n
    N0(i,1)=(A(i,1)+B(i,1)+C(i,1)+D(i,1))*1.0/4;
end

```

```

        dt20(i,1)=sqrt(0.25*((A(i,1)-N(i,1))^2+(B(i,1)-N(i,1))^2+(C(i,1)-
N(i,1))^2+(D(i,1)-N(i,1))^2));
    end
    fenzi=0;fenmu=0;
    for i=1:n
        fenzi=fenzi+dt2(i,1);fenmu=fenmu+dt20(i,1);
    end
    y=fenzi*1.0/fenmu;
end

```

```

function y = f3(n,k,a,b,c,d,t,a1,b1,c1,d1)
    fenzi=0;fenmu=0;
    for j=1:k
        fenmu=fenmu+3*(a1(1,j)+b1(1,j)+c1(1,j)+d1(1,j));
        for i1=1:n-1
            for i2=i1+1:n
                fenzi=fenzi+
(a(i1,j)*a(i2,j)+b(i1,j)*b(i2,j)+c(i1,j)*c(i2,j)+d(i1,j)*d(i2,j))*
(t(i1,1)+t(i2,1))*(t(i1,1)+t(i2,1)-1)*(t(i1,1)+t(i2,1)-2);
            end
        end
    end
    y=fenzi*1.0/fenmu;
end

```

```

function y = g(n,k,A,B,C,D,a,b,c,d,z,m0,a1,b1,c1,d1,t,w,w)
    lambda=[0.8,0.16,0.032,0.008];
    ff=
[ff0(n,k,A,B,C,D,a,b,c,d,z,m0,w,w),f1(n,k,A,B,C,D,a,b,c,d,a1,b1,c1,d1),f2(n,k,a,
b,c,d,A,B,C,D),f3(n,k,a,b,c,d,t,a1,b1,c1,d1)];
    y=dot(ff,lambda);
end

```