

队伍编号	21100140071
题号	D

基于钢材制造业中的钢材切割下料问题的优化方案

摘 要

自 1996 年来我国成为全球第一大钢铁生产国，钢铁行业技术装备水平不断提高，国际竞争力不断增强，钢铁产业前景可期。进入“十四五”时期经济社会发展以推动高质量发展为主题，把发展质量问题摆在更为突出的位置，着力提升发展质量和效益，对钢铁产业的要求不再是数量规模为主，而是品种质量、绿色低碳、创新发展。与此同时，我们应看到钢材制作过程中浪费的原料较多，加工切割模式还有待提高。本文针对钢材制造业中常见的下料问题，利用现有的切割工具，在已满足订单需求的前提下，通过多目标规划、建立数学模型，对不同要求的下料问题给出了更优化可行的切割方案。

针对问题一，我们根据附件1的原料切割方案说明，在不考虑浮动比例的前提下，首先讨论了单张原料用圆盘剪和切头剪可能的剪切方式（见图1），我们选择以产品的宽度作为研究对象，通过对其随机排样，利用 MATLAB 对现有的10种原材料设限，共找出了1000种切割方案（请见附件“10种原料加工成5种卷料的方案表.xls”），根据成材率的定义，对其成材率进行计算并将各原料在不同方案下的成材率进行排序，筛选确定了10种切割方案。最后，通过对两个目标进行指标的标准化，既满足使用原料

张数最少，同时总的成材率尽量高，综合评价，得到原材料最少张数为20张，所用材料编号及材料成材率详见附件结果表单。

针对问题二，我们在问题一已得到的优化方案下增加约束条件，在不考虑浮动比例的前提下，我们选择优先满足卷料订单，利用 MATLAB 首先对卷料订单满足后的余料进行切割优化，为使浪费最小，结合板料订单需求的实际长度，我们将余料进行合理分割成一小块，再对每一小块进行优化切割，以满足板料订单，基于问题一得到原材料使用最少张数为21张。

针对问题三，我们首先利用 MATLAB 运算得到15种订单在10种原料的宽向排布数，得到满足条件的切割方案，并计算了各方案在10种原料上的宽向余料，为使余料最大化利用，选择优先选取单张面积大的原料，用方案中宽余量最小的模式进行切割，完成1次切头剪后，再根据所选方案对应订单是否已经完成判断是否可以执行下一步方案的选取，同时需要判断剩余的原料中的长余量是否小于未完成订单需求长度的最小值，通过反复筛选最终得到原料使用量最小的优化方案，得到最终最优化的结果为：使用了24张原料，换刀次数为43，小机器加工次数为0。

针对问题四，我们考虑到在已指定有浮动比例情况下，订单需求中原长度不同的订单可以实现长度一致化，为此我们首先利用了 MATLAB 计算出全部15种订单中的某两种可能长度一致的可行产品，然后将每组产品在10种原料上的宽余料进行排布，同时计算了宽余量，结合问题三中得到的生产方案，沿用解决问题三的筛选方法，整理得到优化方案，并计算出总成材率。（详细数据请见附件“提交结果.xlsx”）。

关键词：钢材下料 高效切割 整数规划 多目标优化 Matlab Lingo

目录

一、问题重述.....	1
1.1 问题背景.....	1
1.2 需要解决的问题.....	2
二、问题分析.....	3
2.1 概论.....	3
2.2 问题一.....	3
2.3 问题二.....	5
2.4 问题三.....	5
2.5 问题四.....	6
三、模型假设.....	6
四、符号说明.....	7
五、模型建立与求解.....	8
5.1 问题一.....	8
5.2 问题二.....	15
5.2.1 模型建立前的准备.....	15
5.2.2 模型的建立.....	15
5.2.3 模型的求解.....	16
5.3 问题三.....	17
5.3.1 模型建立前的准备.....	17
5.3.2 模型三的建立.....	18
5.3.3 模型三的求解.....	20

5.3.4 模型三的评估.....	21
5.4 问题四.....	23
六、模型评价与推广.....	28
七、参考文献.....	29
八、附录.....	30
8.1 模型程序实现.....	30
8.2 源程序.....	31

一、问题重述

1.1 问题背景

随着全球资源短缺的加剧，钢材价格不断上涨，愈来愈多的企业陷入经营亏损，更多的企业认识到提高钢材利用率的重要性，但是面对钢材切割过程的低利用率和堆积如山的剩余钢材却束手无策。许多企业为降低生产成本、扭亏为盈，采取限制开支甚至减少职工福利等办法，但是收效甚微^[1]。

为此，十分有必要对钢材的生产制造方案进行有效的优化，通过在进行生产前对原材料的切割进行模拟规划，选择最优的切割方案，以期达到既满足需求又节约成本的目的。下料问题普遍存在于玻璃、钢板、木材、纸张、制衣和船舶等行业生产过程中，下料问题按照原材料和零件的维数，分为一维、二维和三维下料问题。其中，一维下料问题指的是型材、棒材的下料，三维下料问题是指原材料和所下零件的长、宽、高均有特定要求的情况^[2]。而二维下料问题则主要指板材的下料，从平面原料上切下各式各样的

下料件，使材料的利用率最高。从计算复杂性理论上，二维下料问题属于具有最高复杂性的一类优化计算问题：NP-完全问题^[3]。

剩余钢材的管理与重复利用，是许多大型企业原材料消耗最为严重的问题。剩余钢材堆积如山，锈蚀流失不计其数，往往把剩余钢材当作废料低价处理，结果，每处理一吨“废钢”，企业就会损失2000多元^[4]。在经济全球化的现代，特别是当下新冠疫情影响全球钢铁行业冷式发展，通过对现有的原材料进行合理可行的切割，在满足订单需求的前提下，尽可能减少生产成本，包括原材料、人力、制造时耗等等，预先确定最优方案，是当下钢材制造业的必由之路。

1.2 需要解决的问题

本文主要利用 MATLAB 对数据进行处理，筛选优化方案，通过构建合理的模型，结合 lingo 运算程序针对不同约束要求进行多目标规划，最终获得原料切割的最优方案，具体问题如下：

问题 1：针对给出的所有原料，请使用最少张数的原材料，满足对5种卷料的要求（不考虑浮动比例），同时尽量提高总的成材率，给出切割方案。

问题 2：针对给出的所有原料，请使用最少张数的原材料，满足对所有订单的要求（不考虑浮动比例），同时尽量提高总的成材率，给出切割方案。

问题 3：圆盘剪每次排刀需要人工更换刀在排刀架上的位置，同时若有材料需要被移到小机器上再次切割也需要人为操作。为减少人力成本，希望尽量减少换刀数和在小机器上切割数。针对给出的所有原料，请使用

最少张数的原材料，满足对所有订单的要求（不考虑浮动比例）。同时尽量减少换刀数和小机器上切割数，并尽量提高总的成材率。给出切割方案。

问题 4：若订单额外指定了浮动比例，则交付的订单长度可在需求长度的基础上上浮。例如，规定浮动比例为5%，则切割出的长度在原有长度的95%到105%之间均满足要求。请重新按照第2问的要求，给出切割方案。

二、问题分析

2.1 概论

以上提出的问题是常见的钢材下料问题，旨在以最小生产成本获得最大综合原料利用率，属于二维的多目标规划，问题的难点在于确定单张钢材最优的切割方式，借助加权求和法，对不同量级指标进行标准化，合理利用余料，对二维矩形件进行排样，构建数学模型。

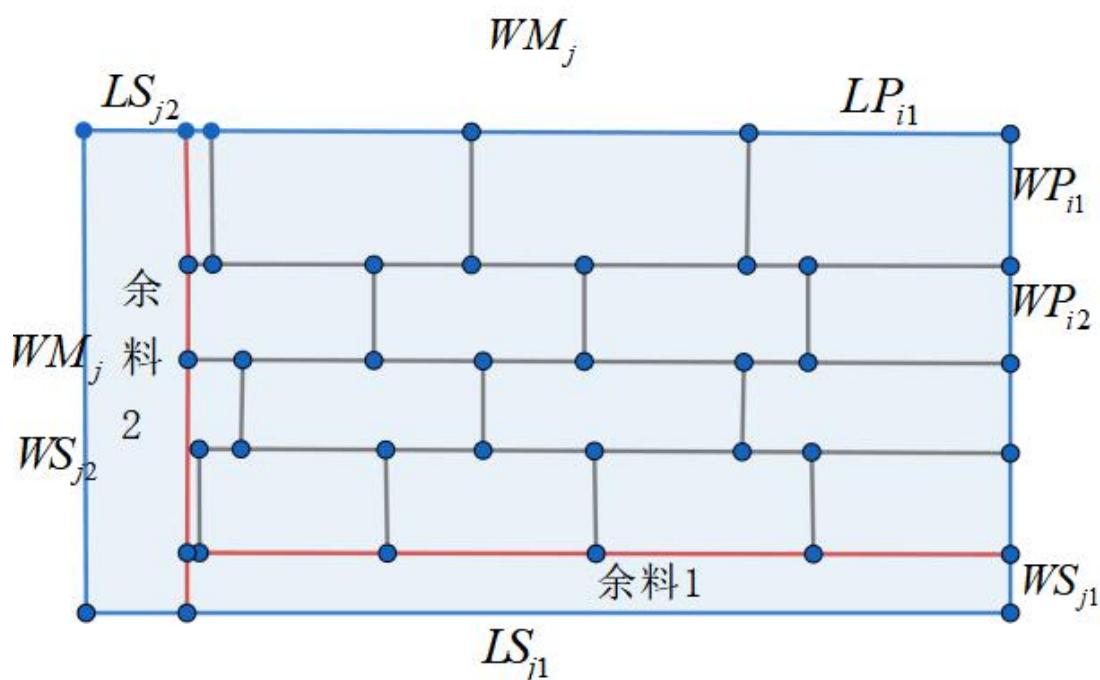
2.2 问题一

剪刀作为剪切设备的重要结构部件，具有确定设备切割范围和切割方式的决定性作用。按剪刀形状与配置特点分为平行刃剪切机、斜刃剪切机和圆盘剪切机^[5]。本题采用圆盘剪和切头剪，对材料进行横向和纵向同时切割，问题的目标为：通过对所有原料的切割方案进行优化，在满足对5种卷料的要求下，使用最少张数的原材料，同时得到尽可能大的成材率。成材率定义：被切割成成品的面积加上可回库的余料面积与使用原料总面积之比。因此本题共有两个目标，即原材料用量最少、成材率尽量高，属于多目标

规划问题。解决此类问题，我们选择分别对两个目标寻求可行方案，最后采取综合评价，通过对结果进行指标标准化，得到最终的优化方案。综合评价是将评价对象在各指标的表现值通过加权得到综合值的一种评价方法。但当多个指标处于不同量级时，直接进行线性组合就会导致权重的失效，所以将指标进行标准化是重要的一步^[6]。

在题一和题二中，我们采取的切割方式如下图，即先用产品的宽对原料的宽边进行一次一维排布，直至宽余料不足以再排列其他任何原料为止。然后，分别对参与第一次排布的每种产品对原料的长边再进行一次一维排布，直至每种产品都不能再排列为止。随后，利用圆盘剪锯切原料至若干种产品排布的最远端，再进行一次切头剪。这样，我们一定能得到两块较为规整的余料，即下图余料 1 和余料 2

图 1 问题一和问题二的切割方式



2.3 问题二

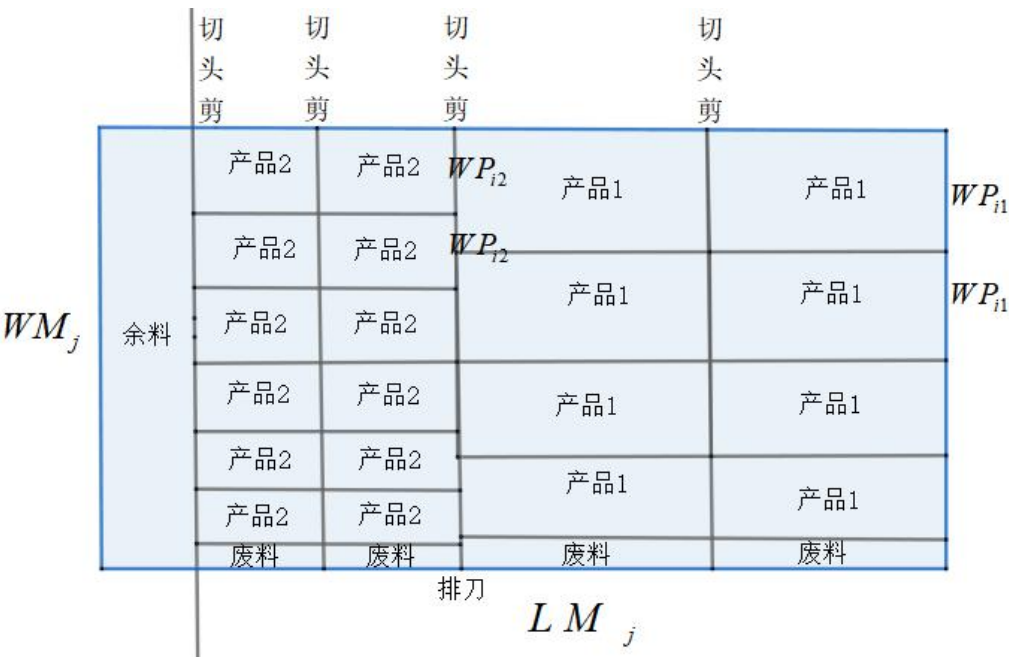
本问是对问题一的进一步延展，一问中我们已经得到在满足卷料订单下所使用最少张数原材料的切割方案以及可回库余料的长宽（见提交结果表），生产过程中每一次使用的板材都可能会出现边角料、残余料，缺乏有效的管理会导致原本可以在残余料中下料的零件被安排到整板中下料，容易导致预料得不到合理的使用，最终被作为废料处理掉，出现“大材小用”、“小材不用”的局面，造成材料浪费^[7]。因此问题一已在满足卷料订单需求的条件下，还需满足板料订单需求，解决问题的关键在于如何合理有效的利用现有剩余的原料，即如何对余料进行剪切，为此我们需要对一问得到的余料分割成小块，根据板料订单的长度选择合适的分割大小，再对每一小块再进行排样。

2.4 问题三

本问涉及排刀次数问题，要求原料使用要最少同时尽量减少人力成本（即圆盘剪的换刀次数和小机器的切割数），得到尽可能高的总成材率。

在这一基础上，我们设计了一种切割方式，将同种在原料上整齐排列，得到一个宽边齐平的一个纵列，这样排列好处在于可以采用切头剪结合圆盘剪直接切割得到产品。详细切割方案如下图：

图 2 针对第三问 节省人力的切割方案



显然，若采用上图这种切割方式，小机器加工的次数必然为 0，已经实现了一个优化目标的最优化。在这基础上，对另外三个优化目标进行再优化，即降低排刀次数、提高成材率、降低消耗的原料张数。需要注意的是，显然采用这种加工方式会产生较多的

废料，尤其是宽向排布的宽余料部分必然会被切成废料。为了降低废料的产生，提高成材率，在选择加工产品时应当使原料的宽余料尽量少。

2.5 问题四

第四问在第三问的基础上指定了浮动比例使订单长度可在需求度的基础上上下浮动。这样一来，使不同长度的产品化为相同长度的产品成为了可能。第四问依然同第三问一样，先用产品的宽对原料的宽边进行一维派样。不同点在于，第三问选择同种产品排样是因为产品的长度不一致，但在添加了浮动比例后，不同的材料长度一致成为了可能，于是可以将长度可一致化的不同种产品共同在原料的宽边上进行排样。这样一来，极大地提高了宽向排布的方案数。

图 3 添加浮比后的一例切割方案



因此，建模之前，需要进行的准备工作是：通过计算给出所有两两之间可以长度一致化的产品，生成一张 15*15 的关系表，记录其一致化后的长度或可继续浮动的长度范围。

随后，将可一致化的两种产品在原料的宽边上共排布，计算给出所有的排布方案，并与第三问得到的简单独立排布方案结合，将所有方案记录后，进行建模。

三、模型假设

1. 假设生产制造过程无机械性损失。
2. 假设加工过程中因大机器加工（圆盘剪，切头剪）次数产生的成本忽略不计。
3. 假设题中所要求的优化指标之一“成材率”意为参与生产的原料的平均成材率，需要对参与生产的原料面积进行加权。
4. 假设题中的“浮动指标”只能对产品的长度进行浮动，不能对产品的宽度进行浮动。

四、符号说明

表 1 符号说明

符 号	说 明
WP_i	第 i 种产品的宽
LP_i	第 i 种产品的长
NWP_i	第 i 种产品宽度可切割的数量
NLP_i	第 i 种产品长度可切割的数量
WM_j	第 j 种原料的宽
LM_j	第 j 种原料的长
NP_j	第 i 种产品的生产数量
NP_{jk}	第 j 种原料采取第 k 种方案切割的数量
RP_{jk}	第 j 种原料采取第 k 种方案切割的成材率
N	使用的原料总张数
R	原料平均成材率

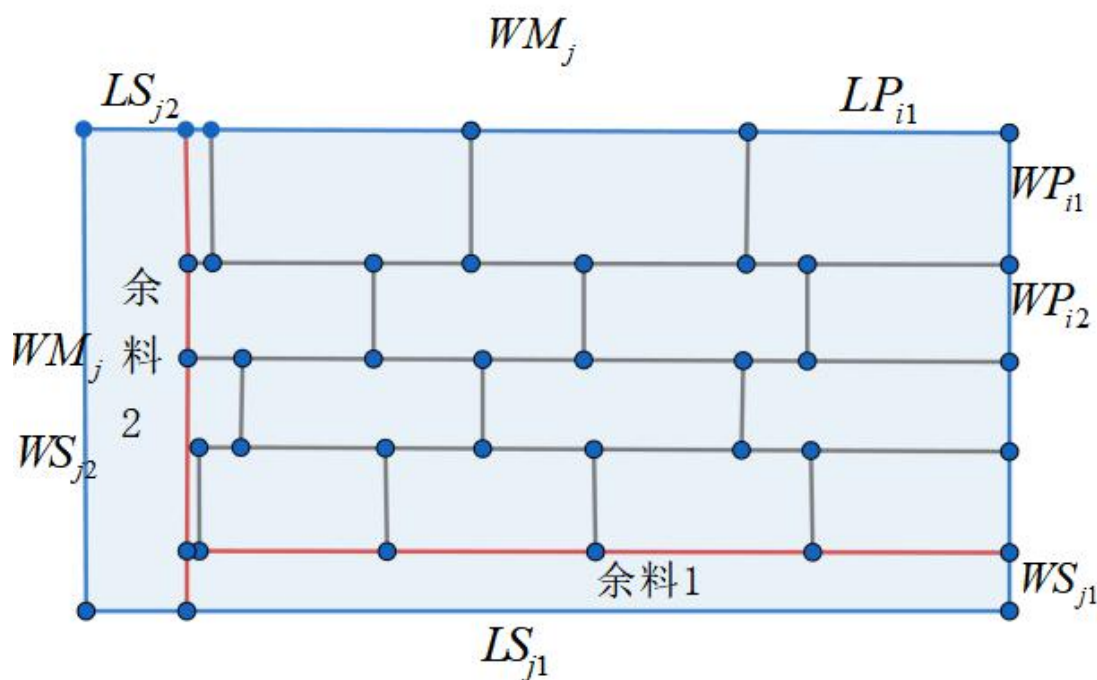
SP_i	原料被切割成产品的数量
SM_j	原料使用总面积
WS_1	可能可回库余料 1 宽度
LS_1	可能可回库余料 1 长度
WS_2	可能可回库余料 2 宽度
LS_2	可能可回库余料 2 长度

五、模型建立与求解

5.1 问题一

5.1.1 模型一建立前的准备

图 4 题一和题二地切割方式



我们首先讨论了单张原料对某种卷料宽度可进行的剪切方式，得到上图结果，根据原材料的特征与空间布局范围将矩形件排样问题化分为三大

类：一维矩形件排样问题、二维矩形件排样问题、以及三维矩形件排样问题^[8-9]，为得到较优的切割模型，我们将问题进行了简化，采用一维线性矩形排样^[10]利用 MATLAB 对每种原料的宽度进行随机排样，10 种原料均得到 1000 种方案，对每种方案对应的成材率进行排序，最终筛选出 10 种较优方案（见表 2）。

表 2 较优切割方案

	订单 1 数量	订单 2 数量	订单 3 数量	订单 4 数量	订单 5 数量	成材率
原料 1	0	0	0	15	0	1
	9	0	0	0	0	1
	0	0	0	0	8	1
	0	0	8	0	2	0.996152028
	3	0	0	9	0	0.994758755
	0	12	2	0	0	0.990363594
	6	0	0	6	0	0.989517509
	0	9	4	0	0	0.980727189
	0	3	0	12	0	0.977251208
	0	0	2	12	0	0.968782479
原料 2	0	0	4	0	0	1
	0	0	0	0	3	1
	0	0	2	0	2	0.97239411
	0	0	3	1	0	0.971638317
	1	0	0	0	2	0.958513808
	0	4	0	0	0	0.957504076
	0	1	3	0	0	0.952700584
	0	0	1	1	2	0.949531755
	0	0	3	0	1	0.939423324
	2	1	0	0	0	0.93501116
原料 3	0	0	0	0	4	1
	0	0	4	0	1	0.99631965
	0	0	3	0	2	0.992639301
	1	0	0	4	0	0.981307819
	3	0	0	1	0	0.979478617
	0	0	4	1	0	0.977652277
	0	0	3	1	1	0.973971927
	0	0	2	0	3	0.970345582
	1	1	0	3	0	0.968788964

	0	1	4	0	0	0.9627302
原	3	0	0	3	0	0.970279468
料	0	0	0	9	0	0.966595797
4	0	3	0	6	0	0.928718133
	3	3	0	0	0	0.923062601
	3	0	2	0	0	0.906377264
	0	0	0	3	2	0.898394274
	0	0	2	6	0	0.895533304
	0	6	0	3	0	0.890840468
	0	3	2	3	0	0.85765564
	0	0	4	3	0	0.824470812
原	0	0	4	0	0	1
料	0	0	0	0	3	1
5	0	0	4	0	0	1
	2	0	0	4	0	0.993244751
	0	0	1	0	2	0.992560736
	0	0	3	0	1	0.974162624
	0	8	0	0	0	0.964398472
	2	2	0	2	0	0.963924667
	2	4	0	0	0	0.950397501
	4	0	0	2	0	0.939145048
原	0	0	4	0	2	0.940955382
料	0	0	0	0	4	0.937631071
6	0	0	2	2	2	0.898524202
	0	0	6	0	0	0.879763551
	2	0	2	0	0	0.872534763
	2	0	0	0	2	0.86063515
	0	2	0	0	2	0.857225819
	0	0	4	2	0	0.837332372
	0	2	4	0	0	0.807845582
	0	0	2	4	0	0.794901192
原	0	0	0	6	0	0.865546191
料	4	0	0	0	0	0.847708264
7	0	2	0	4	0	0.831758958
	2	2	0	0	0	0.819128346
	0	0	0	6	0	0.865546191
	4	0	0	0	0	0.847708264
	0	2	0	4	0	0.831758958
	2	2	0	0	0	0.819128346
	0	0	0	6	0	0.865546191
	4	0	0	0	0	0.847708264

原料8	0	0	2	0	3	0.98741435
	0	0	1	0	4	0.983219133
	0	0	0	1	4	0.978031924
	0	0	4	1	1	0.970330698
	0	0	1	1	3	0.961940265
	4	0	0	0	0	0.959788062
	1	0	3	0	1	0.953203509
	0	0	4	2	0	0.94905183
	2	0	0	3	0	0.945789079
	0	1	0	0	4	0.961087703
原料9	0	0	0	6	0	0.865546191
	4	0	0	0	0	0.847708264
	0	2	0	4	0	0.831758958
	2	2	0	0	0	0.819128346
	0	0	0	6	0	0.865546191
	4	0	0	0	0	0.847708264
	0	2	0	4	0	0.831758958
	2	2	0	0	0	0.819128346
	0	0	0	6	0	0.865546191
	4	0	0	0	0	0.847708264
原料10	0	0	2	0	3	0.98741435
	0	0	1	0	4	0.983219133
	0	0	0	1	4	0.978031924
	0	0	4	1	1	0.970330698
	0	0	1	1	3	0.961940265
	4	0	0	0	0	0.959788062
	1	0	3	0	1	0.953203509
	0	0	4	2	0	0.94905183
	2	0	0	3	0	0.945789079
	0	1	0	0	4	0.961087703

注：根据排样结果知，不必使用原料2和原料4即可得最优解。

5.1.2 模型一的建立

目标函数，希望使用最少张数的原材料同时获得最大成材率。此问为一个多目标规划问题，本文采用加权求和算法^[11]建立模型。由于本题目标函数的两个指标一个需要最小，一个需要最大，故需将两个指标（即使用原料张数，平均成材率）进行标准化处理：

$$n = \frac{N - N_{\min}}{N_{\min}} \quad r = \frac{R_{\max} - R}{R_{\max}}$$

这里的 N_{\min} 为不考虑成材率的情况下使用的原料张数理论最小值， R_{\max} 为不考虑使用张数的情况下平均成材率的理论最大值。通过上述式子计算得到两个标准化后的指标 n 和 r ，若 n 值越小，则实际加工时张数 N 越接近 N_{\min} ；若 r 值越小，则实际加工时平均成材率 R 越接近 R_{\max} 。这样进行标准化的意义在于：可以使得两个指标 n 和 r 可以进行简单的加和，对 $(r+n)$ 求最小值，实现同时对两个目标的优化。

由于无法简单断定这两个指标的重要程度，本文分别对这两个指标赋权 $\frac{1}{2}$ ，得到如下目标函数：

$$\min f = \frac{1}{2} \left(\frac{N - N_{\min}}{N_{\min}} + \frac{R_{\max} - R}{R_{\max}} \right)$$

但，欲求得 N_{\min} 值和 R_{\max} 值，首先得先通过如下两个目标函数来求理论最值。

$$\begin{aligned} \min f &= \sum_{k=1}^{10} \sum_{j=1}^{10} NP_{jk} \\ \max f &= R \end{aligned}$$

三个目标函数共用如下几个约束条件：

约束一：满足5种卷料订单需求：

$$\sum_{j=1}^{10} \sum_{k=1}^{10} NPM_{ijk} NP_{jk} \geq NP_j, i = 1..5$$

约束二：被用来切割的原料数不能超过原料总数：

$$\sum_{k=1}^{10} NP_{jk} \leq NM_j, j = 1, 2..10$$

约束三：原料数量和产品数量均为整数：

$$NP_i, NM_j \in Z$$

约束四：平均成材率 R ，成材率对不同原料的面积进行加权，其计算公式为：

$$R = \frac{\sum_{k=1}^{10} \sum_{j=1}^{10} SM_j NP_{jk} RP_{jk}}{\sum_{k=1}^{10} \sum_{j=1}^{10} SM_j NP_{jk}}$$

约束五：原料2和原料4不满足切割卷料的条件：

$$NP_i = 0, i = 2, 4$$

根据以上目标函数及约束条件，建立以下模型：

$$\begin{aligned} \textcircled{1} N_{\min} &= \min f = \sum_{k=1}^{10} \sum_{j=1}^{10} NP_{jk} \\ \textcircled{2} R_{\max} &= \max f = R \\ \textcircled{3} \min f &= \frac{1}{2} \left(\frac{N - N_{\min}}{N_{\min}} + \frac{R_{\max} - R}{R_{\max}} \right) \end{aligned}$$

$$s.t. \left\{ \begin{array}{l} \sum_{j=1}^{10} \sum_{k=1}^{10} NPM_{ijk} \cdot NP_{jk} \geq NP_j, i = 1..5 \\ \sum_{k=1}^{10} NP_{jk} \leq NM_j, j = 1, 2..10 \\ NP_i, NM_j \in Z \\ R = \frac{\sum_{k=1}^{10} \sum_{j=1}^{10} SM_j NP_{jk} RP_{jk}}{\sum_{k=1}^{10} \sum_{j=1}^{10} SM_j NP_{jk}} \\ NP_i = 0, i = 2, 4 \end{array} \right.$$

5.1.3 模型一的求解

经过 lingo 运算程序，第一步求解得到 N_{\min} 值为20， R_{\max} 值为0.996。

将上述值代入第三个目标函数中，再次求解，得到的最终的优化方案：

表 3 第一问切割方案

材料编号	张数	订单 1	订单 2	订单 3	订单 4	订单 5	成材率
1	2	18	0	0	0	0	1
1	3	18	0	0	18	0	0.9895175
6	1	0	3	2	3	0	0.8576556

6	3	0	18	0	9	0	0.8908405
7	1	0	8	0	0	0	0.9643985
8	8	0	0	32	0	16	0.9409554
10	2	0	0	8	2	2	0.9703307
合计	20	36	29	42	32	18	
需求		36	29	42	32	18	

使用的原料数最少为20张。

5.1.3 模型一的评估

该模型采用的切割方式较为简单，通过一维排样方式先对原料的宽向进行排样，再对长向进行切割。这种切割方式对材料的泛用性较好，且容易得到两块较完整的余料（图1中的余料1和余料2）。但是模型存在缺陷，例如，当产品的长度较大，且原料的长度较小时，容易产生较多废料，造成成材率的降低。再如，该模型忽略了可能产生的余料3：

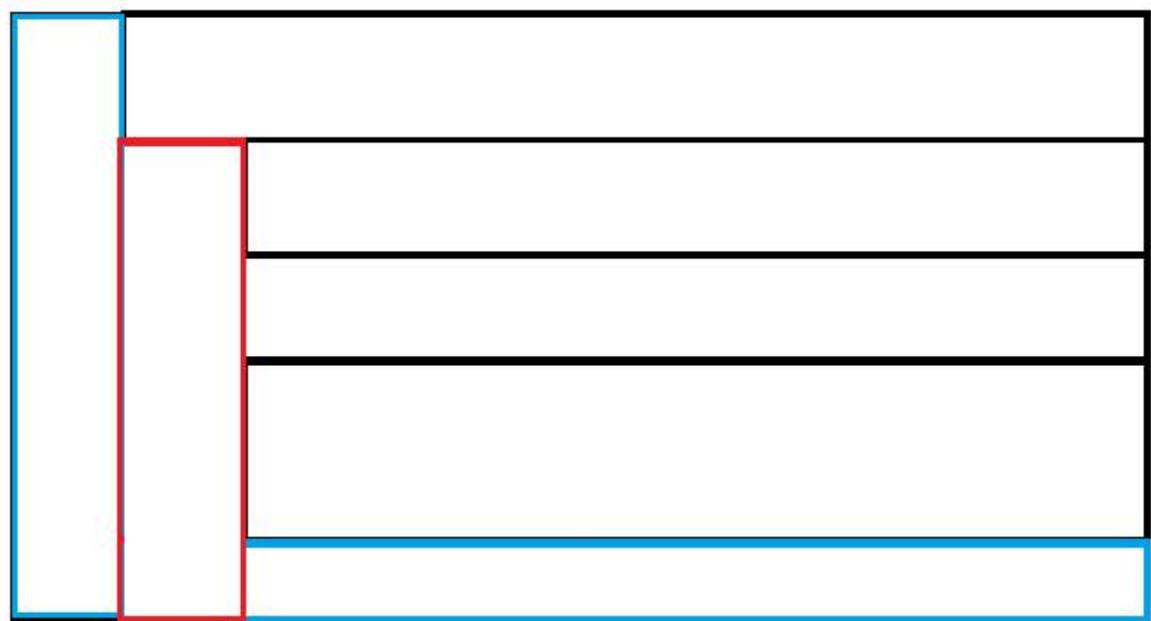


图 5 可能产生的余料 3（红色矩形）

在某些情况下，上图中的红色矩形是有可能成为一个合格的余料的，但是在这个模型中被当作废料处理了。好在所挑选的10种原料各10个切割方案的成材率都较高，发生这种情况的可能较小。

5.2 问题二

5.2.1 模型建立前的准备

在问题一的基础上，根据附件4板料订单中的数据选择合理的长度对余料进行分割，即选择对余料1的长分别以每3000和每4000单独划分为一个版块，而余料2由于宽度不满足生产，故不做排样，经过 matlab 得到划分结果如下所示：

表 4 余料划分方案

材料编号	分割长度	余料宽度	余料分割张数
1	3000	1519.91	22
7	4000	1184.54	5
10	3000	1785.45	2

利用余料虽会造成成材率的降低，但是能够避免使用新的原料。权衡两者，本文认为利用余料更符合生产实际。

可利用的余料量无法满足生产全部的板料订单，在进行多次尝试后，实现了利用余料至少完成了订单 7、8、9、11、12、14。

5.2.2 模型的建立

在问题一的基础上增设约束条件：

约束一追加：满足 10 种板料订单需求：

$$\sum_{j=1}^{10} \sum_{k=1}^{10} NPM_{ijk} \cdot NP_{jk} \geq NP_i, i = 6..15$$

根据以上目标函数及约束条件，建立以下模型：

$$\begin{aligned}
& \textcircled{1} N_{\min} = \min f = \sum_{k=1}^{10} \sum_{j=1}^{10} NP_{jk} \\
& \textcircled{2} R_{\max} = \max f = R \\
& \textcircled{3} \min f = \frac{1}{2} \left(\frac{N - N_{\min}}{N_{\min}} + \frac{R_{\max} - R}{R_{\max}} \right) \\
& s.t. \left\{ \begin{aligned} & \sum_{j=1}^{10} \sum_{k=1}^{10} NPM_{ijk} \cdot NP_{jk} \geq NP_j, i=1..15 \\ & \sum_{k=1}^{10} NP_{jk} \leq NM_j, j=1,2..10 \\ & NP_i, NM_j \in Z \\ & R = \frac{\sum_{k=1}^{10} \sum_{j=1}^{10} SM_j NP_{jk} RP_{jk}}{\sum_{k=1}^{10} \sum_{j=1}^{10} SM_j NP_{jk}} \\ & NP_i = 0, i=2,4 \end{aligned} \right.
\end{aligned}$$

5.2.3 模型的求解

经过 *lingo* 运算程序，得到最终的优化方案：

表 5 题 2 切割方案

材料编号	1	1	6	6	7	8	10	10
张数	2	3	1	3	1	8	2	1
订单 1	18	18	0	0	0	0	0	0
订单 2	0	0	3	18	8	0	0	0
订单 3	0	0	2	0	0	32	8	0
订单 4	0	18	3	9	0	0	2	0
订单 5	0	0	0	0	0	16	2	0
订单 6	0	3	0	0	0	0	0	35
订单 7	24	0	0	0	0	0	0	0
订单 8	16	16	0	0	0	0	0	0
订单 9	24	18	0	0	0	0	0	0
订单 10	0	0	0	0	10	0	6	23

订单 11	6	18	0	0	12	0	0	0
订单 12	0	27	0	0	0	0	0	0
订单 13	4	2	0	0	6	0	4	8
订单 14	8	16	0	0	0	0	0	0
订单 15	0	0	0	0	9	0	4	15

使用的最少原料数为 21 张。（成材率详见附件中的提交结果）

5.3 问题三

5.3.1 模型建立前的准备

建立此问的模型我们需补充符号说明：

表 6 符号说明

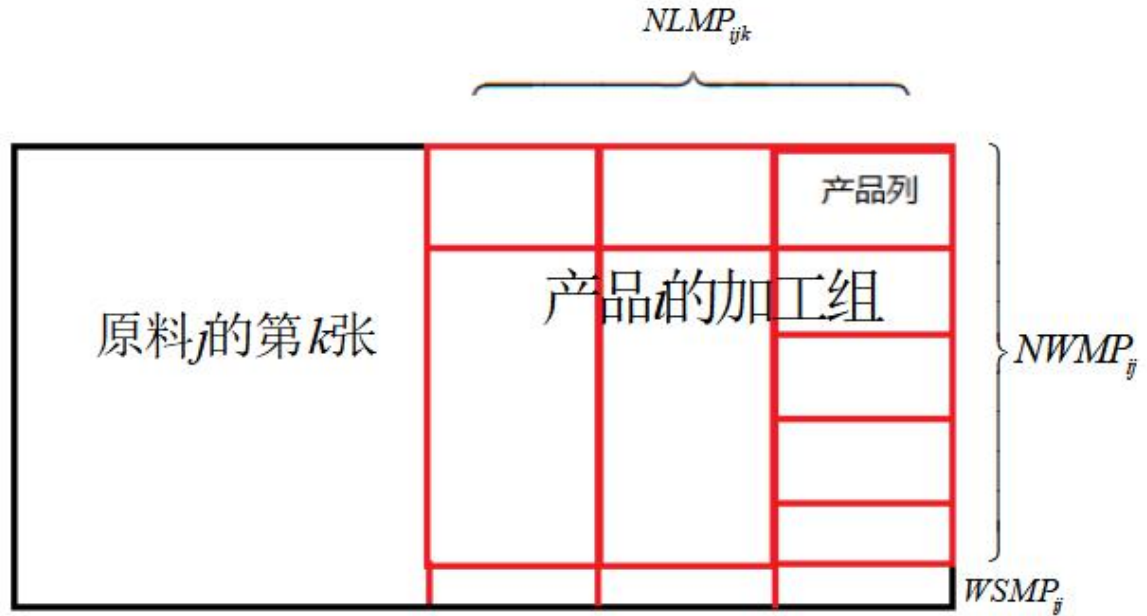
符号	说明
$NWMP_{ij}$	第 i 种产品在第 j 种原料的宽向最大排布数
$NLMP_{ijk}$	第 i 种产品在第 j 种原料的第 k 块上的长向排布数
$WSMP_{ij}$	第 i 种产品在第 j 种原料上切割宽向最大时余宽
N_j	第 j 种原料被切割的张数
C_{jk}	切割第 j 种原料的第 k 张时的换刀次数
LSM_{jk}	第 k 张第 j 种原料的长余量

回顾题三的分析可知，在这一问，为了降低加工人力，我们采取每种原料先对一种产品加工，先在原料的宽边上对产品进行一维排布，排列圆盘剪，开始锯切时，每锯切一列产品就使用切头剪截下一组产品。记录每次使用切头剪后获得的产品数量和原料剩余的长度。若一次使用切头剪后产品满足订单需求，就需要改变一次排刀，再进行第二种产品的加工；若一次使用切头剪后原料的剩余长度不够再生产下一列产品，则需要检索未完成的订单中长度小于原料剩余长度的产品，改变排刀，继续生产。

这种切割方式的好处在于：小机器切割的次数一定为 0，因此，这里的优化指标仅
有三个。

利用 MATLAB 计算出 15 种产品对 10 种原料的宽边最大排布数和宽余料，在加工时，
以一种产品在一种原料上的一列为加工方案，每张原料独立进
行产品的排样。

图 6 加工方案的描述及本题符号的解释



5.3.2 模型三的建立

目标函数，希望使用最少张数的原材料，同时获得最大成材率、最少
的换刀次数以及最少的小机器加工次数。此问在上两问的基础上，增加了
两个优化指标，由于本题采用上述切割方式，故实际只增加了一个优化指
标，即排刀次数。欲采用线性加权和法处理目标函数，故也需将该指标进
行标准化处理。

$$c = \frac{C - C_{\min}}{C_{\min}}$$

目标函数为：

$$\min f = \frac{\left(\frac{N - N_{\min}}{N_{\min}} + \frac{R_{\max} - R}{R} + \frac{C - C_{\min}}{C_{\min}} \right)}{3}$$

考虑到上一问中，我们已经得到了 N_{\min} 值为 21， R_{\max} 值为 0.996，需要先通过以下目标函数求得 C_{\min} ：

$$\min f = C$$

约束一：产品在原料上宽向切割划分的数量应为整数，计算式为：

$$NWMP_{ij} = \left\lfloor \frac{WM_j}{WP_i} \right\rfloor, i = 1..15, j = 1..10$$

注： $\lfloor \cdot \rfloor$ 表示对 $\lfloor \cdot \rfloor$ 内的小数去余处理（向下取整）。

约束二：产品在原料上切割数量应满足订单需求量：

$$\sum_{j=1}^{10} \sum_{k=1}^{NM_j} NLMP_{ijk} NWMP_{ij} \geq NP_i, i = 1, 2..15$$

约束三：参与切割的原料的张数应不超过原料总张数：

$$k \leq NM_j, j = 1..10$$

约束四：参与切割的第 j 种原料总张数计算式：

$$N_j = \sum_{k=1}^{NM_j} \begin{cases} 1, \text{若} \sum_{i=1}^{10} NLMP_{ijk} > 0 \\ 0, \text{若} \sum_{i=1}^{10} NLMP_{ijk} = 0 \end{cases}, j = 1..10$$

约束五：长余量计算式：

$$LSM_{jk} = LM_j - \sum_{i=1}^{15} \sum_{j=1}^{10} \sum_{k=1}^{NM_j} NLMP_{ijk} LP_i$$

约束六：成材率计算式（切割后的余料应满足余料标准）：

$$R = \frac{\sum_{i=1}^{15} \sum_{j=1}^{10} \sum_{k=1}^{NM_j} NLMP_{ijk} NMP_{ij} SP_i + \sum_{j=1}^{10} \sum_{k=1}^{NM_j} \begin{cases} LSM_{jk} WM_j, \text{若} LSM_{jk} \geq 2000, \text{且} WM_j \geq 1000 \\ 0, \text{若} LSM_{jk} < 2000, \text{或} WM_j < 1000 \end{cases}}{\sum_{j=1}^{10} N_j SM_j}$$

约束七：换刀数计算式（每次转换产品或每次转换原料時計 1 次）：

$$C = \sum_{i=1}^{15} \sum_{j=1}^{10} \sum_{k=1}^{NM_j} \begin{cases} 1, NLMP_{ijk} > 0 \\ 0, NLMP_{ijk} \leq 0 \end{cases}$$

根据以上目标函数及约束条件，建立以下模型：

$$\textcircled{1} \min f = C$$

$$\textcircled{2} \min f = \frac{\left(\frac{N - N_{\min}}{N_{\min}} + \frac{R_{\max} - R}{R} + \frac{C - C_{\min}}{C_{\min}} \right)}{3}$$

$$\begin{cases} NWMP_{ij} = \left\lfloor \frac{WM_j}{WP_i} \right\rfloor \\ \sum_{j=1}^{10} \sum_{k=1}^{NM_j} NLMP_{ijk} NWMP_{ij} \geq NP_i \\ R = \frac{\sum_{i=1}^{15} \sum_{j=1}^{10} \sum_{k=1}^{NM_j} NLMP_{ijk} NMP_{ij} SP_i + \sum_{j=1}^{10} \sum_{k=1}^{NM_j} \begin{cases} LSM_{jk} WM_j, \text{若 } LSM_{jk} \geq 2000, \text{ 且 } WM_j \geq 1000 \\ 0, \text{若 } LSM_{jk} < 2000, \text{ 或 } WM_j < 1000 \end{cases}}{\sum_{j=1}^{10} N_j SM_j} \\ s.t. \begin{cases} k \leq NM_j \\ LSM_{jk} = LM_j - \sum_{i=1}^{15} \sum_{j=1}^{10} \sum_{k=1}^{NM_j} NLMP_{ijk} LP_i \\ N_j = \sum_{k=1}^{NM_{j\max}} \begin{cases} 1, \text{若 } \sum_{i=1}^{10} NLMP_{ijk} > 0 \\ 0, \text{若 } \sum_{i=1}^{10} NLMP_{ijk} = 0 \end{cases} \\ C = \sum_{i=1}^{15} \sum_{j=1}^{10} \sum_{k=1}^{NM_j} \begin{cases} 1, NLMP_{ijk} > 0 \\ 0, NLMP_{ijk} \leq 0 \end{cases} \end{cases} \end{cases}$$

5.3.3 模型三的求解

经过 *matlab* 运算程序，得到最终结果：

表 7 第三问切割方案

材料 编号	张数	订单 1	订单 2	订单 3	订单 4	订单 5	订单 6	订单 7	订单 8	订单 9	订单 10	订单 11	订单 12	订单 13	订单 14	订单 15	换刀 次数
1	1	0	15	0	0	0	0	23	0	0	0	0	25	0	22	15	5
1	1	0	10	0	0	4	0	0	0	0	0	16	0	6	0	13	5
1	1	0	0	0	10	4	0	0	0	0	0	0	0	0	0	0	2
1	1	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	1
1	1	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	1
3	1	0	4	0	0	0	38	0	0	40	0	0	0	18	0	0	4

5	1	0	0	0	5	0	0	0	0	0	31	0	0	0	0	0	2
5	1	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	1
6	1	0	0	0	9	0	0	0	0	0	1	2	0	0	0	0	3
6	1	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	1
6	1	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	1
7	1	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	1
8	1	4	0	0	0	0	0	0	3	0	10	16	0	0	0	0	4
8	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
8	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
8	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
8	1	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	1
9	1	4	0	0	0	0	0	0	28	0	0	0	0	0	0	0	2
9	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
10	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
10	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
10	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
10	1	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	1
10	1	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	1

总张数为 24，总换刀次数为 43，总小机器切割次数为 0。（其他数据详见附件“提交结果.xlsx”）

5.3.4 模型三的评估

这种模型对应的切割方式的优劣性十分明显：最大的优点在于小机器加工的次数必然为 0。每次只对同一种产品进行排样，其长边必然相平，可以直接用切头剪进行切割，无需再进行小机器切割，极大地降低了人力要求。

此切割方式的缺点有两处：第一点在于，如果完全采用切头剪，则每块原料的下方宽余料必然会被完全切成废料。第二点在于，由于在宽边排样得到的值一定是个正整数，则在进行了长向切割后，得到的产品数量一定是这个正整数的倍数，这样很容易超量生产。例如，一种产品的需求量为 25，宽向排布数为 3，则在进行了 9 次切割后，得到了 27 个产品，会超量生产 2 个产品，而这两个产品也会成为废料。该模型的这两个缺点都会导致成材率的降低。

图 7 产生废料示意图

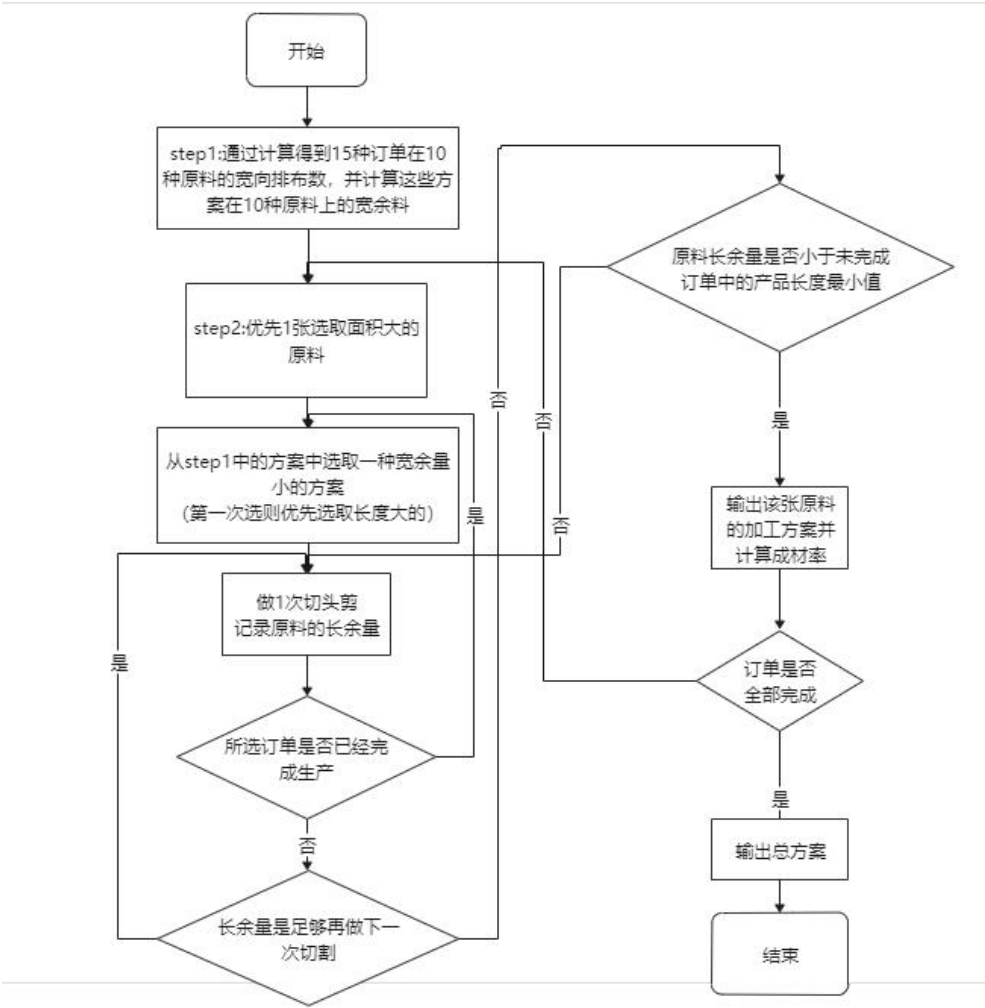


不过好在：算法在选择需要生产的产品时，会优先选择宽向排布时宽余量小的产品进行生产，这样计算，第一个缺点就得到了一定程度的改善。

并且，在对本问采用的 *matlab* 编程中，采用随机尝试的方式寻找更优解，可以在极短的时间内进行百万次计算，很容易找到高成材率的加工方案。

5.3.5 程序实现流程示意图

图 8 题 3 程序实现流程图



5.4 问题四

5.4.1 模型建立前的准备

本题在问题三的基础上，切割方案由一次对一种产品的切割一列，扩充至对可以长度浮动一致化的若干产品进行组合排样得到的全部方案。利用 MATLAB 计算得到所有的切割方案，记录为 a_j （原料 j 的所有单列切割方案）（详见附件“plan.mat”）

建立本问的模型，我们需要补充符号说明：

表 8 符号说明 3

符 号	说 明
$NLMQ_{a_jjk}$	第 k 张第 j 种原料采用第 a_j

LMQ_{a_jj}	种切割方案的长向排布数
$NMPQ_{ija_j}$	第 j 种原料的第 a_j 种切割方案的单列长度
NQ_j	第 j 种原料采用第 a 种切割方案切割一次得到的产品 i 的数量
a_j	第 j 种原料的切割方案总数
	j 种原料的第 a 种加工方案

5.4.2 模型四的建立

目标函数，希望在增设浮动比例下使用最少原料张数同时获得最大总成材率、最少的换刀次数：

$$\min f = \frac{(\frac{N - N_{\min}}{N_{\min}} + \frac{R_{\max} - R}{R} + \frac{C - C_{\min}}{C_{\min}})}{3}$$

约束一：选择的原料加工方案数应不超过原料切割方案数：

$$a_j \leq NQ_j, a \in Z, j = 1..10$$

约束二：切割量应满足需求：

$$\sum_{i=1}^{15} \sum_{j=1}^{10} \sum_{a_j=1}^{NQ_j} \sum_{k=1}^{NM_j} NMPQ_{ija_j} NLMQ_{a_jjk} \geq NP_i$$

约束三：加工量不允许超额供应：

$$k \leq NM_j, j = 1..10$$

约束四：切割方案排布总长应不超过原料总长：

$$\sum_{a_j=1}^{NQ_j} \sum_{k=1}^{NM_j} LMQ_{a_jj} NLMQ_{a_jjk} \leq LM_j$$

约束五：原料切割的总张数应为产品在原料上长向切割数量和：

$$N_j = \sum_{k=1}^{NM_j} \begin{cases} 1, \text{若} \sum_{j=1}^{10} NLMQ_{a_jjk} > 0 \\ 0, \text{若} \sum_{j=1}^{10} NLMQ_{a_jjk} = 0 \end{cases}$$

约束六：长余量计算式：

$$LSM_{jk} = LM_j - \sum_{a_j=1}^{NQ_j} \sum_{j=1}^{10} \sum_{k=1}^{NM_j} NLMQ_{a_jjk} LMQ_{a_jj}$$

约束七：成材率的计算式：

$$R = \frac{\sum_{a_j=1}^{NQ_j} \sum_{j=1}^{10} \sum_{k=1}^{NM_j} NLMQ_{a_jjk} NMPQ_{ija_j} SP_i + \sum_{j=1}^{10} \sum_{k=1}^{NM_j} \begin{cases} LSM_{jk} WM_j, \text{若} LSM_{jk} \geq 2000, \text{且} WM_j \geq 1000 \\ 0, \text{若} LSM_{jk} < 2000, \text{或} WM_j < 1000 \end{cases}}{\sum_{j=1}^{10} N_j SM_j}$$

约束八：换刀次数计算式（每次更换切割方案，或每次更换原料计 1 次）：

$$C = \sum_{a_j=1}^{NQ_j} \sum_{j=1}^{10} \sum_{k=1}^{NM_j} \begin{cases} 1, NLMQ_{a_jjk} > 0 \\ 0, NLMQ_{a_jjk} = 0 \end{cases}$$

建立如下模型：

$$\min f = \frac{(\frac{N - N_{\min}}{N_{\min}} + \frac{R_{\max} - R}{R} + \frac{C - C_{\min}}{C_{\min}})}{3}$$

$$s.t. \begin{cases} a_j \leq NQ_j, a \in Z \\ \sum_{i=1}^{15} \sum_{j=1}^{10} \sum_{a_j=1}^{NQ_j} \sum_{k=1}^{NM_j} NMPQ_{ija_j} NLMQ_{a_jjk} \geq NP_i \\ R = \frac{\sum_{a_j=1}^{NQ_j} \sum_{j=1}^{10} \sum_{k=1}^{NM_j} NLMQ_{a_jjk} NMPQ_{ija_j} SP_i + \sum_{j=1}^{10} \sum_{k=1}^{NM_j} \begin{cases} LSM_{jk} WM_j, \text{若 } LSM_{jk} \geq 2000, \text{ 且 } WM_j \geq 1000 \\ 0, \text{若 } LSM_{jk} < 2000, \text{ 或 } WM_j < 1000 \end{cases}}{\sum_{j=1}^{10} N_j SM_j} \\ k \leq NM_j \\ \sum_{a_j=1}^{NQ_j} \sum_{k=1}^{NM_j} LMQ_{a_jj} NLMQ_{a_jjk} \leq LM_j \\ N_j = \sum_{k=1}^{NM_j} \begin{cases} 1, \text{若 } \sum_{j=1}^{10} NLMQ_{a_jjk} > 0 \\ 0, \text{若 } \sum_{j=1}^{10} NLMQ_{a_jjk} = 0 \end{cases} \\ LSM_{jk} = LM_j - \sum_{a_j=1}^{NQ_j} \sum_{j=1}^{10} \sum_{k=1}^{NM_j} NLMQ_{a_jjk} LMQ_{a_jj} \\ C = \sum_{a_j=1}^{NQ_j} \sum_{j=1}^{10} \sum_{k=1}^{NM_j} \begin{cases} 1, NLMQ_{a_jjk} > 0 \\ 0, NLMQ_{a_jjk} = 0 \end{cases} \end{cases}$$

5.4.3 模型四的求解

经 *matlab* 编程运算，得到最终结果：

表 9 第四问的切割方案

材料 编号	张数	订单 1	订单 2	订单 3	订单 4	订单 5	订单 6	订单 7	订单 8	订单 9	订单 10	订单 11	订单 12	订单 13	订单 14	订单 15
1	1	0	0	6	0	4	0	0	0	0	36	15	13	0	0	28
1	1	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	6	0	0	6	0	0	0	0	0	0	0	0	0	0	0
1	1	6	0	0	6	0	0	0	0	0	0	0	0	0	0	0
1	1	4	0	3	3	0	0	0	0	0	0	0	0	0	0	0
3	1	0	0	2	0	2	38	0	0	0	0	0	6	0	0	0
5	1	0	0	2	0	3	0	6	24	0	0	0	0	24	22	0
6	1	0	0	6	0	0	0	11	0	7	0	16	0	0	0	0
6	1	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0

7	1	0	4	3	0	1	0	0	0	15	0	0	0	0	0	0
7	1	0	4	4	0	0	0	0	0	0	6	0	0	0	0	0
7	1	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0
7	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	0	0	4	0	2	0	0	7	0	0	0	0	0	0	0
9	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	1	0	0	4	0	2	0	3	0	9	0	0	0	0	0	0
10	1	0	0	4	0	2	0	3	0	9	0	0	0	0	0	0
10	1	0	0	4	0	2	0	0	0	0	0	3	6	0	0	0
10	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0

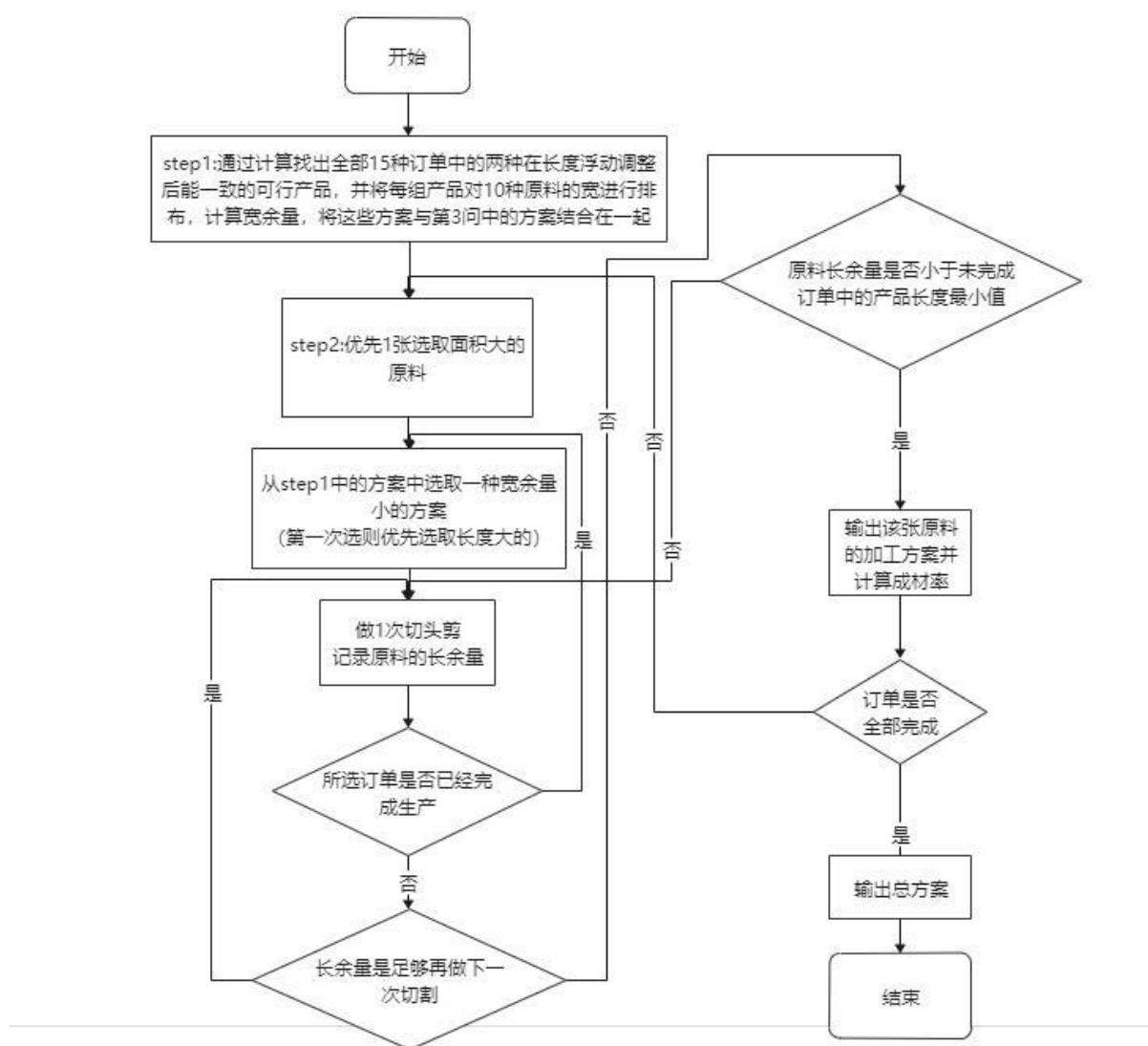
总张数为 22，总换刀次数为 43，小机器加工次数为 0。（详细数据见附件”提交结果.xlsx”）

5.4.4 模型的评估

本问的模型沿用了第三问的模型，优缺点与第三问的模型相仿。小机器加工次数仍然为 0。对第三问的模型最大的改善就是将可浮动长度的原料等长化，这样做很大程度上增加了宽向排样的方案量，更有利于选择一种宽余量更小的排样方案。在算法的设计过程中，程序自动检测某个生产订单是否完成，会自行删除包含已完成订单的加工方案，这样做，也极大地避免了超量生产这一缺点，综合来看，是在第三问的模型上的一种升级。

5.4.5 程序实现的流程示意图

图 9 题 4 程序实现流程图



六、模型评价与推广

本文对钢铁加工是切割下料问题进行了建模分析。题 1、2 中建立的模型能够有效地降低消耗的原料数、提高成材率，但是切割方案不合实际：小机器加工次数过大，对人力要求太高。不同于 1、2，在 3、4 两问中建立的数学模型采用了纯切头剪切割的加工方式，在极大减小人力要求的情况下，对成材率和耗材张数进行再优化，此模型更具有实用意义。

也许，在此模型能够应用到钢铁长加工的规划中。设计合适的程序和智能系统，为钢铁加工厂提供实时的加工方案指导，能极大提高钢铁厂生产钢产品的效率且极大地降低生产花费。

七、参考文献

- [1] 李浩. 钢材切割现状及 6478 切割软件的应用[J].电焊机.第 35 卷第 1 期.20-23.2015.
- [2] 张克,林家恒.二维下料问题的研究进展[R].山东大学学报(工学版).第 35 卷增刊.129-132.2005.
- [3] GAREY M R,JOHNSON D S.Computer and intractability guide to the theory of NP-completeness[M].San Francisco;W H Freeman Company.1979.
- [4] 陈新会. 现代切割技术在数控切割中的应用[J].机械管理开发.第 3 期 (总第 90 期).51-52.2006.
- [5] 黄庆学.轧钢机械设计[M].冶金工业出版社.46-49.2007.
- [6] 刘竞妍,张可,王桂华. 综合评价中数据标准化方法比较研究[R].数字技术与应用.第 36 卷第 06 期.84-85.2018.
- [7] 张圣,李继.智能算法的板材下料优化排样系统研究 [J] .机械与电子.(4)10-13.2010.
- [8] 贾志欣.排样问题的研究现状与趋势[J].计算机辅助设计与图形学学报.16(7):890-897.2004.
- [9] 陈钊.求解矩形件排问题的离散粒子群算法[D].武汉: 华中科技大学.2017.
- [10] 邓文浩. 满足“一刀切”约束的矩形件排样问题研究[D].辽宁.大连海事大学.2019.

- [11] 杨 满.汪传旭.徐 朗. 基于加权求和算法的多目标订货决策模型[J].
工业工程. 第 19 卷第 5 期.2016.

八、附录

8.1 模型程序实现

8.1.1 模型一的实现

第一问的程序实现：

第一步：使用 matlab 运行 Question1_Plan_Pick.m，得到输出值 Plan，为 10 张 1000*6 的加工方案表，分别对应 10 种原料的 1000 种加工方案。

每张表的每一行为该原料的一种加工方案，第六列为该加工方案的成材率。

第二步：10 张表导入到 excel 中，分别把 10 张表对成材率作降序排列，删除重复加工方案，挑选 10 中成材率较高的方案，这 10 组 10*5 的加工方案和对应的 10 组 10*1 的成材率数据导出到文本文件中。

第三步：使用 lingo 打开 Question1.lg4，将 10 组加工方案和成材率导入到 lingo 的数据段中，写入约束条件，进行求解。

8.1.2 模型二的实现

第二问的程序实现：

第一步：使用 matlab 运行 Question2_Plan_Pick.m，得到输出值 Plan，为 10 张 100*11 的加工方案表，分别对 10 种原料对长度分小后对 10 种板料的 100 种加工方案。

第七列为该加工方案的成材率。

第二步：将其中的 1，7，10 各挑 10 种优选方案，用 lingo 编程，先将 1，7，10 的余料尝试加工产品 7，8，9，11，12，14。

第三步：选取一块新的原料，加工剩余订单。

8.1.3 模型三的实现

第三问程序实现：

第一步：使用 matlab 运行 Question3_Step1.m，得到名为 num_kind_width 和 ws1_m_p 的两个变量，分别对应模型中的 $NWMP_{ij}$ 和 $WSMP_{ij}$ （产品在原料宽向上的最大排布数，和最大排布时的宽余量）。

第二步：使用 matlab 运行 Question3_Step2.m，随机排样切割，累计排刀次数，通过程序最上方 n 值的可以设置此程序的迭代次数。循环计算完成后，输出名为 DIALOG.mat 的变量，为该程序计算得到的最优加工方案。

8.1.4 模型四的实现

第一步：使用 matlab 运行 Question4_Prepare.m，得到名为 Canjoin 的变量，为 15*15 的矩阵，表示通过调整浮比后是否可以实现等长度的产品。若值为 0，表示两种产品无法等长。若值为一个常数或一个范围，表示两种产品可以实现等长。值为范围的两种产品的长度可以通过调整浮比控制在这一范围内。

第二步：使用 matlab 运行 Question4_Step1.m，得到名为 plan 的变量，为 10*15 的一个元胞数组，每一个元胞显示该种原料包含该种产品的所有加工方案（长度为 5 的一个数组），供加工时选择方案。该长度为 5 的数组位置一表示第一种产品的序号，位置二表示第一种产品的宽向排列数，位置三表示第二种产品的序号，位置四表示第二种产品的宽向排列数，位置五表示该排样方案对该材料的宽余量，位置六表示浮比最短长度，位置七表示浮比最长长度。

第三步：使用 matlab 运行 Question4_Step2.m，随机排样切割，累计排刀次数，计算方案中加工材料的浮比。通过程序最上方 n 值的可以设置此程序的迭代次数。循环计算完成后，输出名为 DIALOG.mat 的变量，为该程序计算得到的最优加工方案。

8.2 源程序

第 1 问使用的 matlab 以及 lingo 程序

Question1_Prepare.m

```
function PLAN=Question1_Prepare()  
  
product=xlsread('附件 3_4.xlsx','订单','B2:G16');  
  
material=xlsread('附件 3_4.xlsx','原料','B2:E11');  
  
PLAN=cell(10,6);
```

```

plan=zeros(10,10);
for j=1:10
    if material(j,1)>min(product(1:5,1))
        i_list=find(product(1:5,1)<material(j,1));

        ws1=material(j,2); %用于计算宽度余量

        ls1=material(j,1); %用于计算长度余量

        num_kind_width=zeros(1,5);
        num_kind_length=zeros(1,5);
        while ws1>min(product(1:5,2))

            i=randchoose(i_list,1);%随机从5种产品中选一个
            if ws1-product(i,2)<0
                continue
            end
            ws1=ws1-product(i,2);
            num_kind_width(i)=num_kind_width(i)+1;
            num_kind_length(i)=fix(material(j,1)/product(i,1));
            num_kind_width_length=[num_kind_width,
num_kind_length];
        end
        for k=1:10
            plan(j,k)=num_kind_width_length(k);
        end

        ws2=material(j,1)-max(plan(j,6:10).*product(1:5,1)');
        ls2=material(j,2);
        ls1=ls1-ws2;

        %长度余量

        S_Product=0;
        for n=1:5

            S_Product=S_Product+plan(j,n)*plan(j,n+5)*product(n,6);
        end
        S_surplus=0;
        if ws1>100 && ls1>50000
            S_surplus=S_surplus+ws1*ls1;
        end
        if ws2>2000 && ls2>1000
            S_surplus=S_surplus+ws2*ls2;

```

```

        end
        R=(S_surplus+S_Product)/material(j,4);

        %写入上述计算结果

        PLAN{j,6}=R;
        for n=1:5

PLAN{j,n}=[num_kind_width_length(n),num_kind_width_length(n+5)
];
        end
    end
end
end

```

Question1_Plan_Pick.m

```

clear
clc
Plan_M=cell(10,1);

for n=1:1000
    PLAN=Question1_Prepare();
    for i=1:length(PLAN(:,1))
        Planj=cell(1,length(PLAN(1,:)));
        if any(PLAN{i,6})
            for j=1:length(PLAN(1,:))
                Planj{1,j}=PLAN{i,j};
            end
        end
        Plan_M{i,1}=[Plan_M{i,1};Planj];
    end
end

```

Question1.lg4

```

model:

sets:

material/1..10/:wm,lm,nm,Sm;

product/1..5/:wc,lc,nc;

plan/1..10/;

link1(material,plan,product):ncm;

link2(material,plan):np,p;

endsets

```

data:

wm=1519.91 999.35 1232.32 920.62 1573.71 844.99 1184.54 879.02 969.02 1785.45;

lm=148623.91 32960.49 75508.72 14091.52 75040.31 138570.39 98641.28 114074.27
104637.72 58023.82;

nm=5 10 8 2 3 10 12 9 3 10;

Sm=225894967 32939065.68 93050905.83 12972935.14 118091686.3 117090593.8
116844541.8 100273564.8 101396043.4 103598629.4;

wc=422.88 282.88 268.36 277.7 332.29;

lc=44351.13 39229.01 54787.74 45284.39 53479.79;

nc=36 29 42 32 18;

ncm=

0 0 0 15 0

9 0 0 0 0

0 0 0 0 8

0 0 8 0 2

3 0 0 9 0

0 12 2 0 0

6 0 0 6 0

0 9 4 0 0

0 3 0 12 0

3 3 0 6 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0	0	0	0	0
---	---	---	---	---

0	0	0	0	0
---	---	---	---	---

0	0	4	0	0
---	---	---	---	---

0	0	0	0	3
---	---	---	---	---

0	0	2	0	2
---	---	---	---	---

0	0	3	1	0
---	---	---	---	---

1	0	0	0	2
---	---	---	---	---

0	4	0	0	0
---	---	---	---	---

0	1	3	0	0
---	---	---	---	---

0	0	1	1	2
---	---	---	---	---

0	0	0	0	0
---	---	---	---	---

0	0	0	0	0
---	---	---	---	---

0	0	0	0	0
---	---	---	---	---

0	0	0	0	0
---	---	---	---	---

0	0	0	0	0
---	---	---	---	---

0	0	0	0	0
---	---	---	---	---

0	0	0	0	0
---	---	---	---	---

0	0	0	0	0
---	---	---	---	---

0	0	0	0	0
---	---	---	---	---

0	0	0	0	0
---	---	---	---	---

0	0	0	0	0
---	---	---	---	---

0	0	0	0	0
---	---	---	---	---

0	0	0	0	4
---	---	---	---	---

0	0	4	0	1
---	---	---	---	---

0	0	3	0	2
---	---	---	---	---

1	0	0	4	0
---	---	---	---	---

3	0	0	1	0
---	---	---	---	---

0	0	4	1	0
0	0	3	1	1
0	0	2	0	3
1	1	0	3	0
0	1	4	0	0

3	0	0	3	0
0	0	0	9	0
0	3	0	6	0
3	3	0	0	0
3	0	2	0	0
0	0	0	3	2
0	0	2	6	0
0	6	0	3	0
0	0	0	0	0
0	0	0	0	0

0	0	4	0	0
0	0	0	0	3
0	0	4	0	0
2	0	0	4	0
0	0	1	0	2
0	0	3	0	1
0	8	0	0	0
2	2	0	2	0
2	4	0	0	0
4	0	0	2	0

0	0	4	0	2
0	0	0	0	4

0	0	2	2	2
0	0	6	0	0
2	0	2	0	0
2	0	0	0	2
0	2	0	0	2
0	0	4	2	0
0	2	4	0	0
0	0	0	0	0

0	0	0	6	0
4	0	0	0	0
0	2	0	4	0
2	2	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

0	0	2	0	3
0	0	1	0	4
0	0	0	1	4
0	0	4	1	1
0	0	1	1	3
4	0	0	0	0
1	0	3	0	1
0	0	4	2	0
2	0	0	3	0

0 1 0 0 4;

p=

1

1

1

0.996152028

0.994758755

0.990363594

0.989517509

0.980727189

0.977251208

0.972009962

0 0 0 0 0 0 0 0 0 0

1

1

0.97239411

0.971638317

0.958513808

0.957504076

0.952700584

0.949531755

0

0

0 0 0 0 0 0 0 0 0 0

1

0.99631965
0.992639301
0.981307819
0.979478617
0.977652277
0.973971927
0.970345582
0.968788964
0.9627302

0.970279468
0.966595797
0.928718133
0.923062601
0.906377264
0.898394274
0.895533304
0.890840468
0
0

1
1
1
0.993244751
0.992560736
0.974162624
0.964398472
0.963924667
0.950397501

0.939145048

0.940955382

0.937631071

0.898524202

0.879763551

0.872534763

0.86063515

0.857225819

0.837332372

0.807845582

0

0.865546191

0.847708264

0.831758958

0.819128346

0

0

0

0

0

0

0.98741435

0.983219133

0.978031924

0.970330698

0.961940265

0.959788062

```

0.953203509

0.94905183

0.945789079

0.961087703

;

min_num_m=21;

max_rate_m=0.996244;

enddata


submodel min_num:
min=@sum(link2:np);

@for(product(i):@sum(link2(j,k):np(j,k)*ncm(j,k,i))>nc(i));

@for(material(j):@sum(plan(k):np(j,k))<nm(j));

@for(link2:@gin(np));

endsubmodel


submodel max_rate:

max=@sum(link2(j,k):p(j,k)*Sm(j)*np(j,k))/@sum(link2(j,k):np(j,k)*Sm(j));

@for(product(i):@sum(link2(j,k):np(j,k)*ncm(j,k,i))>nc(i));

@for(material(j):@sum(plan(k):np(j,k))<nm(j));

@for(link2:@gin(np));

endsubmodel


min=n+r;

r=(max_rate_m-(@sum(link2(j,k):p(j,k)*Sm(j)*np(j,k))/@sum(link2(j,k):np(j,k)
)*Sm(j))))/max_rate_m/2;

```

```

n=(@sum(link2:np)-min_num_m)/2/min_num_m;

@for(product(i):@sum(link2(j,k):np(j,k)*ncm(j,k,i))>nc(i));

@for(material(j):@sum(plan(k):np(j,k))<nm(j));

@for(link2:@gin(np));

end

```

第二问使用的 matlab 以及 lingo 程序

Question2_Prepare.m

```

function PLAN=Question2_Prepare()

product=xlsread('附件 3_4.xlsx','订单','B2:G16');

material=xlsread('附件 3_4.xlsx','原料','B2:E11');

PLAN=cell(10,11);

plan=zeros(10,20);
for j=1:10
    if material(j,1)>min(product(6:15,1))
        i_list=find(product(6:15,1)<material(j,1));

        ws1=material(j,2); %用于计算宽度余量

        if j==1
            ls1=3000;
        elseif j==7
            ls1=4000;
        else
            ls1=3000; %用于计算长度余量
        end

        num_kind_width=zeros(1,10);
        num_kind_length=zeros(1,10);
        while ws1>min(product(i_list+5,2))

            i=randchoose(i_list,1);%随机从 5 种产品中选一个

            if ws1-product(i+5,2)<0
                continue
            end
        end
    end
end

```

```

        ws1=ws1-product(i+5,2);
        num_kind_width(i)=num_kind_width(i)+1;
        num_kind_length(i)=fix(ls1/product(i+5,1));
        num_kind_width_length=[num_kind_width,
num_kind_length];
    end
    for k=1:20
        plan(j,k)=num_kind_width_length(k);
    end

    ws2=ls1-max(plan(j,11:20).*product(6:15,1)');
    ls2=material(j,2);
    ls1=ls1-ws2;

    %长度余量

    S_Product=0;
    for n=1:10

S_Product=S_Product+plan(j,n)*plan(j,n+10)*product(n+5,6);
    end
    S_surplus=0;
    if ws1>100 && ls1>50000
        S_surplus=S_surplus+ws1*ls1;
    end
    if ws2>2000 && ls2>1000
        S_surplus=S_surplus+ws2*ls2;
    end
    R=(S_surplus+S_Product)/material(j,2)/ls1;

    %写入上述计算结果

    PLAN{j,11}=R;
    for n=1:10

PLAN{j,n}=[num_kind_width_length(n),num_kind_width_length(n+10
)];
    end
    end
end
end

```

Question2_Plan_Pick.m

```
clear
clc
Plan_M=cell(10,1);
tic
for n=1:100
    PLAN=Question2_Prepare();
    for i=1:length(PLAN(:,1))
        Planj=cell(1,length(PLAN(1,:)));
        if any(PLAN{i,1})
            for j=1:length(PLAN(1,:))
                Planj{1,j}=PLAN{i,j};
            end
        end
        Plan_M{i,1}=[Plan_M{i,1};Planj];
    end
end
toc
```

Question2.lg4

model:

sets:

material/1..3/:nm;

product/1..6/:nc;

plan/1..10/;

link1(material,plan,product):ncm;

link2(material,plan):np;

endsets

data:

nm=22 5 1;

nc=23

31

40

34

25

22

;

ncm=

5	4	0	0	0	0
0	0	5	5	0	8
10	0	0	0	0	0
0	0	0	0	10	8
0	0	0	0	5	8
0	0	0	0	10	0
0	0	0	5	0	8
0	4	0	0	0	8
0	4	5	0	0	0
0	0	0	5	5	0

0	0	5	0	0	8
0	0	5	0	0	0
0	0	5	0	5	0
0	4	5	0	0	0
0	0	0	0	0	8
0	4	0	0	0	0
0	0	0	0	0	0
0	0	0	0	5	0
5	0	0	0	0	0
0	0	10	0	0	0

0	8	0	0	0	0
---	---	---	---	---	---

```
0  0  5  0  0  8
0  0  5  0  0  0
0  0  5  0  5  0
0  4  5  0  0  0
0  4  5  0  0  0
0  0  0  0  0  8
0  4  0  0  0  0
0  0  0  0  0  0
0  0  0  0  5  0
;
```

```
enddata
```

```
submodel min_num:
```

```
min=@sum(link2:np);
```

```
@for(product(i):@sum(link2(j,k):np(j,k)*ncm(j,k,i))>nc(i));
```

```
@for(material(j):@sum(plan(k):np(j,k))<nm(j));
```

```
@for(link2:@gin(np));
```

```
endsubmodel
```

```
calc:
```

```
@solve(min_num);
```

```
endcalc
```

第三问使用的 matlab 程序:

Question3_Step1.m

```
clear
clc

product=xlsread('附件 3_4.xlsx','订单','B2:G16');

material=xlsread('附件 3_4.xlsx','原料','B2:E11');

num_kind_width=zeros(10,15);
ws1_m_p=zeros(10,15);

%% 记录所有产品在原料的宽向排布最大值和余料宽度
for j=1:10
    for i=1:15
        ws1=material(j,2); %用于计算宽度余量
        while ws1-product(i,2)>0
            ws1=ws1-product(i,2);
            num_kind_width(j,i)=num_kind_width(j,i)+1;
        end
        ws1_m_p(j,i)=ws1;
    end
end

%% 记录卷料在原料的长向排布最大值和余料长度
ls1_m_p=zeros(10,5);
num_kind_length=zeros(10,5);
for j=1:10
    for i=1:5
        ls1=material(j,1); %用于计算宽度余量
```

```

        while ls1-product(i,1)>0
            ls1=ls1-product(i,1);
            num_kind_length(j,i)=num_kind_length(j,i)+1;
        end
        ls1_m_p(j,i)=ls1;
    end
end

%% 记录该产品在原料宽向排布最大的情况下长向排布的总长度

len_p_total=zeros(10,15);
for j=1:10
    for i=1:15
        num_len=fix(product(i,3)/num_kind_width(j,i))+1;
        len_p_total(j,i)=num_len*product(i,1);

    end
end

```

Question3_Step2.m

```

tic
MIN_F=100;

Nmin=21;%完成加工任务所需的最小张数

Rmax=1;%最大成材率

Cmax=66;%经 100000 次随机计算得到的最大排刀次数

Cmin=37;%经 100000 次随机计算得到的最小排刀次数

for n=1:150000000 %设置程序的运行次数

    %% 开始安排切割工作

    %记录任务状态，1 为待完成，0 为已完成
    Mission=ones(length(product(:,3)),1);
    Mission_num=product(:,3);

    %记录原料状态，1 为有余，0 为用尽
    Material=ones(length(material(:,3)),1);
    Material_num=material(:,3);

```

%加工日志的建立

```
dialog=cell(1,length(Mission)+7);  
dialog{1}='所选原料序号';  
for i=1:length(Mission)  
    dialog{i+1}=['订单',num2str(i)];  
end  
dialog{length(Mission)+2}='成材率';  
dialog{length(Mission)+3}='换刀次数';  
dialog{length(Mission)+4}='余料 1 长';  
dialog{length(Mission)+5}='余料 1 宽';  
dialog{length(Mission)+6}='余料 2 长';  
dialog{length(Mission)+7}='余料 2 宽';
```

% 为了实现张数最小，先从大面积原料开始切割，为了提高成材率，挑选宽度>1000，且宽度越大越好的材料

```
while any(Mission) %任务全部完成时，结束循环
```

```
    c_change=0;%记录排刀次数
```

```
    dialog1=cell(1,length(Mission)+7);  
    dialog2=zeros(1,15);
```

```
    S=[];%列出可选择的原材料的面积
```

```
    for i=1:length(Material)  
        if Material(i)==1  
            S=[S,material(i,4)];  
        else  
            S=[S,0];  
        end  
    end
```

```
    %随机打乱优先选择原料的顺序，多次排布
```

```
    Choose=[10,1,6,2,9,5,8,4,3,7];  
    Choose=randchoose(Choose,10);
```

```

for i=1:length(Material)
    if Material(i)==0
        Choose(i)=0;
    end
end

%进行下一张原材料选择

num_M=find(Choose==max(Choose),1);
dialog1{1}=num_M;

ls1=material(num_M,1);%原料初始长度


%进行第一次加工产品选择。优先选择宽余量小的卷料，再后选择宽余量小的板料

pro_choose=inf(15,1);
for i=1:length(Mission)
    if Mission(i)==1
        pro_choose(i)=ws1_m_p(num_M,i);
    end
end
if any(Mission(1:5,1))
    num_P=find(pro_choose(1:5,1)==min(pro_choose(1:5,1)),1);
else
num_P=find(pro_choose(6:15,1)==min(pro_choose(6:15,1)),1)+5;
end

%开始加工产品,每使用一次切头剪记录一次剩余原料长度


num_numP_len=0; %该加工产品在原料长向的个数
if ls1-product(num_P,1)>0
    c_change=c_change+1;%首次排刀
end
while ls1-product(num_P,1)>0

Mission_num(num_P,1)=Mission_num(num_P,1)-num_kind_width(num_M
,num_P);
    ls1=ls1-product(num_P,1);
    num_numP_len=num_numP_len+1;
    if Mission_num(num_P,1)<=0
        Mission(num_P,1)=0;
        pro_choose(num_P,1)=inf;

```

```

        break
    end
end
if Mission(num_P)==1
    dialog2(num_P)=num_numP_len*num_kind_width(num_M,num_P);
else

dialog2(num_P)=num_numP_len*num_kind_width(num_M,num_P)+Mission_num(num_P,1);
    end

    %进行第 n 次加工产品的选择, 选择板料或卷料, 也要求宽余量尽量小
    while min(pro_choose)<0.5*material(num_M,2)

        %先找取卷料
        num_P=0;
        for i=1:5
            if Mission(i)==1 &&
pro_choose(i)<0.5*material(num_M,2) && product(i,1)<ls1
                num_P=i;
            end
        end
        %卷料不行, 选择板料
        if num_P==0

num_P=find(pro_choose(6:15,1)==min(pro_choose(6:15,1)),1)+5;
            end

            if pro_choose(num_P)==inf%若板料也全都加工完了, 结束
                break
            end

            num_numP_len=0;%该加工产品在原料长向的个数
            if ls1-product(num_P,1)>0
                c_change=c_change+1;%先排刀
            end
            while ls1-product(num_P,1)>0

Mission_num(num_P,1)=Mission_num(num_P,1)-num_kind_width(num_M
,num_P);
                ls1=ls1-product(num_P,1);

```

```

        num_numP_len=num_numP_len+1;
        if Mission_num(num_P,1)<=0
            Mission(num_P,1)=0;
            pro_choose(num_P,1)=inf;
            break
        end
    end

    if Mission(num_P)==1

dialog2(num_P)=num_numP_len*num_kind_width(num_M,num_P);
        else

dialog2(num_P)=num_numP_len*num_kind_width(num_M,num_P)+Mission_num(num_P,1);
        end

        %建立停止对该材料加工的条件, 若满足, 结束循环, 进入下一次加工

        if all(Mission(6:15,1))
            break
        end

        if Mission(num_P)==0
            continue
        else
            pro_choose(num_P)=inf;
        end

    end

    %计算成材率
    s_p=0;
    for i=1:15
        s_p=s_p+dialog2(i)*product(i,6);
    end
    if ls1>2000 && material(num_M,2)>1000
        s_p=s_p+material(num_M,2)*ls1;
    end
    r=s_p/material(num_M,4);

    %写入日志
    dialog1{length(Mission)+2}=r;
    dialog1{length(Mission)+3}=c_change;

```

```

    for i=1:15
        dialog1{i+1}=dialog2(i);
    end
    if ls1>2000 && material(num_M,2)>1000
        dialog1{length(Mission)+4}=ls1;
        dialog1{length(Mission)+5}=material(num_M,2);
    end
    dialog=[dialog;dialog1];

    %原料状态更新

    Material_num(num_M)=Material_num(num_M)-1;
    if Material_num(num_M)==0
        Material(num_M)=0;
    end
end

%优化目标的处理

%总张数

num_total_m=length(dialog(:,1))-1;
N=(num_total_m-Nmin)/Nmin;

%最终成材率

s_useful=0;
s_total=0;
for i=2:length(dialog(:,1))

    s_useful=s_useful+material(dialog{i},4)*dialog{i,length(Mission)+2};
    s_total=s_total+material(dialog{i},4);
end
r_final=s_useful/s_total;
R=(Rmax-r_final)/Rmax;

%最终换刀次数

C_Change=0;
for i=2:length(dialog(:,1))
    C_Change=C_Change+dialog{i,length(Mission)+3};
end
C=(C_Change-Cmin)/(Cmax-Cmin);

%目标函数

minf=(N+C+R)/3;

```

```
if minf<MIN_F
    MIN_F=minf;
    DIALOG=dialog;
    C_CHANGE=c_change;
end
end
toc
```

第四问使用的 matlab 程序:

Question4_Prepare.m

%% 长度同一化处理

```
material=xlsread('附件 3_4.xlsx','原料','B2:E11');

product=xlsread('附件 3_4.xlsx','订单','B2:G16');

CanJoin=cell(15,15);
for i=1:15
    for j=1:15
        imin=product(i,1);
        imax=product(i,1);
        jmin=product(j,1);
        jmax=product(j,1);
        if product(i,4)>0
            imin=imin*(1-product(i,4));
            imax=imax*(1+product(i,4));
        end
        if product(j,4)>0
            jmin=jmin*(1-product(j,4));
            jmax=jmax*(1+product(j,4));
        end
        if imin<jmax && jmin<imax && imin~=imax && jmin~=jmax
            CanJoin{i,j}=[max([imin,jmin]),min(imax,jmax)];
        elseif imin<jmax && jmin<imax && imin==imax && jmin~=jmax
            CanJoin{i,j}=[imin,imax];
        elseif imin<jmax && jmin<imax && imin~=imax && jmin==jmax
            CanJoin{i,j}=[jmin,jmax];
        else
            CanJoin{i,j}=[0,0];
        end
    end
end
end
```

```
product=xlsread('附件 3_4.xlsx','订单','B2:G16');

material=xlsread('附件 3_4.xlsx','原料','B2:E11');

num_kind_width=zeros(10,15);
ws1_m_p=zeros(10,15);
num_kind_width_canjoin=cell(10,15);

%% 记录所有产品在原料的宽向排布最大值和余料宽度

for j=1:10
    for i=1:15
        for k=i:15
            if CanJoin{i,k}~=0
                ws1_list=[];
                numi_list=[];
                numj_list=[];
                num_i=fix(material(j,2)/product(i,2));
                num_j=0;
                ws1=material(j,2)-num_i*product(i,2);
                ws1_list=[ws1_list,ws1];
                numi_list=[numi_list,num_i];
                numj_list=[numj_list,num_j];
                while num_i~=0
                    num_i=num_i-1;
                    ws1=material(j,2)-num_i*product(i,2);
                    num_j=fix(ws1/product(k,2));
                    ws1=ws1-num_j*product(k,2);
                    ws1_list=[ws1_list,ws1];
                    numi_list=[numi_list,num_i];
                    numj_list=[numj_list,num_j];
                end
                for l=1:length(ws1_list)
                    if i~=k && numi_list(l)~=0 && numj_list(l)~=0
                        num_kind_width_canjoin{j,i}=[num_kind_width_canjoin{j,i};i,num
                        i_list(l),k,numj_list(l),ws1_list(l),CanJoin{i,k}];
                    end
                end
            end
        end
    end
end
```

```

end

    wsl=material(j,2); %用于计算宽度余量
    while wsl-product(i,2)>0
        wsl=wsl-product(i,2);
        num_kind_width(j,i)=num_kind_width(j,i)+1;
    end
    wsl_m_p(j,i)=wsl;
end
end

%两种决策方案的整合
plan=cell(10,15);
for i=1:15
    for j=1:10

plan{j,i}=[num_kind_width_canjoin{j,i};i,num_kind_width(j,i),0
,0,wsl_m_p(j,i),product(i,1),product(i,1)];
    end
end

```

Question4_Step2.m

```

tic
MIN_F=100;

Nmin=21;%完成加工任务所需的最小张数

Rmax=1;%最大成材率

Cmax=66;%最大排刀次数

Cmin=37;%最小排刀次数

for n=1:10000000

%% 开始安排切割工作

%记录任务状态，1 为待完成，0 为已完成
Mission=ones(length(product(:,3)),1);
Mission_num=product(:,3);

```

%记录原料状态，1 为有余，0 为用尽

```
Material=ones(length(material(:,3)),1);  
Material_num=material(:,3);
```

%加工日志的建立

```
dialog=cell(1,length(Mission)+14);  
dialog{1}='所选原料序号';  
for i=1:length(Mission)  
    dialog{i+1}=['订单',num2str(i)];  
end  
dialog{length(Mission)+2}='成材率';  
dialog{length(Mission)+3}='换刀次数';  
dialog{length(Mission)+4}='余料 1 长';  
dialog{length(Mission)+5}='余料 1 宽';  
dialog{length(Mission)+6}='余料 2 长';  
dialog{length(Mission)+7}='余料 2 宽';  
dialog{length(Mission)+8}='订单 1 浮比';  
dialog{length(Mission)+9}='订单 4 浮比';  
dialog{length(Mission)+10}='订单 5 浮比';  
dialog{length(Mission)+11}='订单 7 浮比';  
dialog{length(Mission)+12}='订单 8 浮比';  
dialog{length(Mission)+13}='订单 12 浮比';  
dialog{length(Mission)+14}='订单 15 浮比';  
%随机挑选进行加工的材料
```

```

while any(Mission) %任务全部完成时，结束循环

    C_change=0; %记录换刀次数

    dialog1=cell(1,length(Mission)+14);
    dialog2=zeros(1,15);

    S=[];%列出可选择的原材料的面积
    for i=1:length(Material)
        if Material(i)==1
            S=[S,material(i,4)];
        else
            S=[S,0];
        end
    end
    %随机打乱优先选择原料的顺序，多次排布
    Choose=[10,1,6,2,9,5,8,4,3,7];
    Choose=randchoose(Choose,10);
    for i=1:length(Material)
        if Material(i)==0
            Choose(i)=0;
        end
    end
    %进行下一张原材料选择
    num_M=find(Choose==max(Choose),1);
    dialog1{1}=num_M;

    lsl=material(num_M,1);%原料初始长度

    if any(Mission(6:15,1)) %如果板料还没加工完，先加工卷料，板料作为补
短
        %进行加工产品方案选择
        plan_choose=[];
        for i=1:length(Mission)
            for j=1:length(plan{num_M,i}(:,1))
                if plan{num_M,i}(j,3)==0
                    if Mission(plan{num_M,i}(j,1))==1

```

```

plan_choose=[plan_choose;plan{num_M,i}(j,:)];
    end
    else
        if Mission(plan{num_M,i}(j,1))==1 &&
Mission(plan{num_M,i}(j,3))==1

plan_choose=[plan_choose;plan{num_M,i}(j,:)];
    end
    end

    end
end

    %将一系列方案先按宽余量升序排列（减小浪费），再按长度降序排列（先生产
长度较大的产品）
    plan_choose=sortrows(plan_choose,5);
    plan_choose=sortrows(plan_choose,7,'descend');

    %选取最优方案
    plan_choose1=plan_choose(1,:);

    %开始加工产品,每使用一次切头剪记录一次剩余原料长度

    num_numP_len=0; %该加工方案在原料长向的个数
    if ls1-plan_choose1(1,6)>0
        C_change=C_change+1; %一次排刀
    end

    while ls1-plan_choose1(1,6)>0 %为了减小张数，优先采用浮动长度
的最小值进行生产

Mission_num(plan_choose1(1,1),1)=Mission_num(plan_choose1(1,1)
,1)-plan_choose1(1,2);
        if plan_choose1(1,3)~=0

Mission_num(plan_choose1(1,3),1)=Mission_num(plan_choose1(1,3)
,1)-plan_choose1(1,4);
        end

```

```

ls1=ls1-plan_choose1(1,6);%优先剪浮动长度的最小值
num_numP_len=num_numP_len+1;

if plan_choose1(1,3)~=0
    if Mission_num(plan_choose1(1,1),1)<=0 ||
Mission_num(plan_choose1(1,3),1)<=0
        if Mission_num(plan_choose1(1,1),1)<=0 &&
Mission_num(plan_choose1(1,3),1)<=0
            Mission(plan_choose1(1,1),1)=0;
            Mission(plan_choose1(1,3),1)=0;
        elseif Mission_num(plan_choose1(1,1),1)<=0
            Mission(plan_choose1(1,1),1)=0;
        else
            Mission(plan_choose1(1,3),1)=0;
        end

        break
    end
else
    if Mission_num(plan_choose1(1,1),1)<=0
        Mission(plan_choose1(1,1),1)=0;
        break
    end
end
end

if plan_choose1(1,3)~=0
    %浮比计算

    if plan_choose(1,1)==1

dialog1{length(Mission)+8}=plan_choose1(1,6)/product(1,1);
        elseif plan_choose(1,1)==4

dialog1{length(Mission)+9}=plan_choose1(1,6)/product(4,1);
        elseif plan_choose(1,1)==5

dialog1{length(Mission)+10}=plan_choose1(1,6)/product(5,1);
        elseif plan_choose(1,1)==7

dialog1{length(Mission)+11}=plan_choose1(1,6)/product(7,1);
        elseif plan_choose(1,1)==8

dialog1{length(Mission)+12}=plan_choose1(1,6)/product(8,1);

```

```

elseif plan_choose(1,1)==12

dialog1{length(Mission)+13}=plan_choose1(1,6)/product(12,1);
    elseif plan_choose(1,1)==15

dialog1{length(Mission)+14}=plan_choose1(1,6)/product(15,1);
    end
    if plan_choose(1,3)==1

dialog1{length(Mission)+8}=plan_choose1(1,6)/product(1,1);
    elseif plan_choose(1,3)==4

dialog1{length(Mission)+9}=plan_choose1(1,6)/product(4,1);
    elseif plan_choose(1,3)==5

dialog1{length(Mission)+10}=plan_choose1(1,6)/product(5,1);
    elseif plan_choose(1,3)==7

dialog1{length(Mission)+11}=plan_choose1(1,6)/product(7,1);
    elseif plan_choose(1,3)==8

dialog1{length(Mission)+12}=plan_choose1(1,6)/product(8,1);
    elseif plan_choose(1,3)==12

dialog1{length(Mission)+13}=plan_choose1(1,6)/product(12,1);
    elseif plan_choose(1,3)==15

dialog1{length(Mission)+14}=plan_choose1(1,6)/product(15,1);
    end

%浮比计算

%产品数计算

if Mission(plan_choose1(1,1),1)==1

dialog2(plan_choose1(1,1))=dialog2(plan_choose1(1,1))+num_numP
_len*plan_choose1(1,2);
    else

dialog2(plan_choose1(1,1))=dialog2(plan_choose1(1,1))+num_numP
_len*plan_choose1(1,2)+Mission_num(plan_choose1(1,1));
    end
    if Mission(plan_choose1(1,3),1)==1

```

```

dialog2(plan_choose1(1,3))=dialog2(plan_choose1(1,3))+num_numP
_len*plan_choose1(1,4);
    else

dialog2(plan_choose1(1,3))=dialog2(plan_choose1(1,3))+num_numP
_len*plan_choose1(1,4)+Mission_num(plan_choose1(1,3));
    end
    else
        if Mission(plan_choose1(1,1),1)==1

dialog2(plan_choose1(1,1))=dialog2(plan_choose1(1,1))+num_numP
_len*plan_choose1(1,2);
        else

dialog2(plan_choose1(1,1))=dialog2(plan_choose1(1,1))+num_numP
_len*plan_choose1(1,2)+Mission_num(plan_choose1(1,1));
        end
    end
end
end

    %进行第 n 次加工产品的选择, 选择板料或卷料, 也要求宽余量尽量小
while 1

    %进行加工产品方案选择(删除其中包含已完成订单的方案)
    plan_choose=[];
    for i=1:length(Mission)
        for j=1:length(plan{num_M,i}(:,1))
            if plan{num_M,i}(j,3)==0
                if Mission(plan{num_M,i}(j,1))==1

plan_choose=[plan_choose;plan{num_M,i}(j,:)];
                    end
                else
                    if Mission(plan{num_M,i}(j,1))==1 &&
Mission(plan{num_M,i}(j,3))==1

plan_choose=[plan_choose;plan{num_M,i}(j,:)];
                        end
                    end
                end
            end
        end
    end
end
end

```

 %将一系列方案先按宽余量升序排列（减小浪费），此时不再考虑产品长度。（因为此时是要么板料已经加工完，要么材料只有剩料的情况）

```
    plan_choose=sortrows(plan_choose,5);
```

 %选取最优方案

```
    plan_choose1=plan_choose(1,:);
```

 %若无法加工，选下一个

```
    for i=2:length(plan_choose(:,1))
        if ls1<plan_choose1(1,6)
            plan_choose1=plan_choose(i,:);
        else
            break
        end
    end
```

 %若仍然无法加工，直接打破循环

```
    if ls1<plan_choose1(1,6)
        break
    end
```

 %开始加工产品,每使用一次切头剪记录一次剩余原料长度

```
    num_numP_len=0; %该加工方案在原料长向的个数
```

```
    if ls1-plan_choose1(1,6)>0
```

```
        C_change=C_change+1; %一次排刀
```

```
    end
```

 while ls1-plan_choose1(1,6)>0 %为了减小张数，优先采用浮动长度的最小值进行生产

```
    Mission_num(plan_choose1(1,1),1)=Mission_num(plan_choose1(1,1),1)-plan_choose1(1,2);
```

```
        if plan_choose1(1,3)~=0
```

```
    Mission_num(plan_choose1(1,3),1)=Mission_num(plan_choose1(1,3),1)-plan_choose1(1,4);
```

```
    end
```

```

ls1=ls1-plan_choose1(1,6);%优先剪浮动长度的最小值
num_numP_len=num_numP_len+1;

if plan_choose1(1,3)~=0

    if Mission_num(plan_choose1(1,1),1)<=0 ||
Mission_num(plan_choose1(1,3),1)<=0
        if Mission_num(plan_choose1(1,1),1)<=0 &&
Mission_num(plan_choose1(1,3),1)<=0
            Mission(plan_choose1(1,1),1)=0;
            Mission(plan_choose1(1,3),1)=0;
        elseif Mission_num(plan_choose1(1,1),1)<=0
            Mission(plan_choose1(1,1),1)=0;
        else
            Mission(plan_choose1(1,3),1)=0;
        end

        break
    end
else
    if Mission_num(plan_choose1(1,1),1)<=0
        Mission(plan_choose1(1,1),1)=0;
        break
    end
end
end

if plan_choose1(1,3)~=0
    %浮比计算

    if plan_choose(1,1)==1

dialog1{length(Mission)+8}=plan_choose1(1,6)/product(1,1);
        elseif plan_choose(1,1)==4

dialog1{length(Mission)+9}=plan_choose1(1,6)/product(4,1);
        elseif plan_choose(1,1)==5

dialog1{length(Mission)+10}=plan_choose1(1,6)/product(5,1);
        elseif plan_choose(1,1)==7

dialog1{length(Mission)+11}=plan_choose1(1,6)/product(7,1);

```

```

elseif plan_choose(1,1)==8

dialog1{length(Mission)+12}=plan_choose1(1,6)/product(8,1);
    elseif plan_choose(1,1)==12

dialog1{length(Mission)+13}=plan_choose1(1,6)/product(12,1);
    elseif plan_choose(1,1)==15

dialog1{length(Mission)+14}=plan_choose1(1,6)/product(15,1);
    end
    if plan_choose(1,3)==1

dialog1{length(Mission)+8}=plan_choose1(1,6)/product(1,1);
    elseif plan_choose(1,3)==4

dialog1{length(Mission)+9}=plan_choose1(1,6)/product(4,1);
    elseif plan_choose(1,3)==5

dialog1{length(Mission)+10}=plan_choose1(1,6)/product(5,1);
    elseif plan_choose(1,3)==7

dialog1{length(Mission)+11}=plan_choose1(1,6)/product(7,1);
    elseif plan_choose(1,3)==8

dialog1{length(Mission)+12}=plan_choose1(1,6)/product(8,1);
    elseif plan_choose(1,3)==12

dialog1{length(Mission)+13}=plan_choose1(1,6)/product(12,1);
    elseif plan_choose(1,3)==15

dialog1{length(Mission)+14}=plan_choose1(1,6)/product(15,1);
    end

%浮比计算

%产品数计算

if Mission(plan_choose1(1,1),1)==1

dialog2(plan_choose1(1,1))=dialog2(plan_choose1(1,1))+num_numP
_len*plan_choose1(1,2);
    else

dialog2(plan_choose1(1,1))=dialog2(plan_choose1(1,1))+num_numP

```

```

_len*plan_choose1(1,2)+Mission_num(plan_choose1(1,1));
    end
    if Mission(plan_choose1(1,3),1)==1

dialog2(plan_choose1(1,3))=dialog2(plan_choose1(1,3))+num_numP
_len*plan_choose1(1,4);
    else

dialog2(plan_choose1(1,3))=dialog2(plan_choose1(1,3))+num_numP
_len*plan_choose1(1,4)+Mission_num(plan_choose1(1,3));
    end
    else
        if Mission(plan_choose1(1,1),1)==1

dialog2(plan_choose1(1,1))=dialog2(plan_choose1(1,1))+num_numP
_len*plan_choose1(1,2);
        else

dialog2(plan_choose1(1,1))=dialog2(plan_choose1(1,1))+num_numP
_len*plan_choose1(1,2)+Mission_num(plan_choose1(1,1));
        end
    end
end

%建立循环结束条件
if any(Mission)
    continue
else
    break
end

end

%计算成材率
s_p=0;
for i=1:15
    s_p=s_p+dialog2(i)*product(i,6);
end
if ls1>2000 && material(num_M,2)>1000
    s_p=s_p+material(num_M,2)*ls1;
end
r=s_p/material(num_M,4);

```

```

%写入日志

dialog1{length(Mission)+2}=r;
dialog1{length(Mission)+3}=C_change;
for i=1:15
    dialog1{i+1}=dialog2(i);
end
if ls1>2000 && material(num_M,2)>1000
    dialog1{length(Mission)+4}=ls1;
    dialog1{length(Mission)+5}=material(num_M,2);
end

m_p=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
for i=2:length(Mission)+1
    m_p(i-1)=dialog1{i};
end

dialog=[dialog;dialog1];

%原料状态更新

Material_num(num_M)=Material_num(num_M)-1;
if Material_num(num_M)==0
    Material(num_M)=0;
end
end

%优化目标的处理

%总张数

num_total_m=length(dialog(:,1))-1;
N=(num_total_m-Nmin)/Nmin;

%最终成材率

s_useful=0;
s_total=0;
for i=2:length(dialog(:,1))

s_useful=s_useful+material(dialog{i},4)*dialog{i,length(Mission)+2};
    s_total=s_total+material(dialog{i},4);
end
r_final=s_useful/s_total;
R=(Rmax-r_final)/Rmax;

```

```
%最终换刀次数为 C_change1
C_change1=0;
for i=2:length(dialog(:,1))
    C_change1=C_change1+dialog{i,length(Mission)+3};
end
C=(C_change1-Cmin)/(Cmax-Cmin);

%目标函数
minf=(N+C+R)/3;

if C_change1<MIN_F
    MIN_F=C_change1;
    DIALOG=dialog;
    C_CHANGE=C_change1;
end
end
toc
```
