中国研究生创新实践系列大赛

中国光谷·“华为杯”第十九届中国研究生数学建模竞赛

学　　校		淮阴工学院
参赛队号		B22110490001
队员姓名	1.	邵明振
	2.	仇宏扬
	3.	蒋令杰

中国研究生创新实践系列大赛

中国光谷·“华为杯”第十九届中国研究生数学建模竞赛

题 目　　方形件组批优化问题

摘　　要：

在中国制造 2025 目标背景之下，发展环境保护型、资源节约型的智能制造业已成为制造行业的当务之急。为了应对客户提出的各式各样的产品需求、订单组批难且产品质量要求高的问题，使用数学模型辅助企业对定制化产品进行组批优化具有重要意义。本文通过构建**整数线性规划规划模型**来对方形件组批问题进行优化，使其不仅实现了较好的板材原片利用率，同时还对订单组批问题实现了优化。具体做法如下：

针对子问题 1：建立了板材原片的切割方式、产品项(item)、栈(stack)、条带(stripe)、产品位置、生产订单需求等众多因素的一个**全局 0-1 线性规划模型**，以板材原片使用数量最少作为目标函数。在建立模型的过程中，参考了文献【3】中的**整数线性规划模型**，但该模型存在着产品位置不可旋转的问题，板材利用率有待提高。为此，本文所建立的模型中着重考虑了产品可以旋转的情况，极大的提高了板材原片的利用率。该模型变量复杂度较高，计算量大。在编写算法过程中，通过本文建立的模型对子问题 1 中的四个数据表格进行排样优化。在优化之前，本文通过 matlab 对 dataA1~dataA4 的数据进行预处理，计算出每个数据理论上最少的板材原片数量，分别是：**84、83、84、82**。通过本文建立的全局 **0-1 线性规划模型**进行优化，得出优化后的板材原片使用数量分别是：**89、88、88、87**，dataA1~dataA4 的板材原片利用率分别是：**94.382%、94.318%、95.455%、94.253%**，其中四个数据表中**单块板材原片利用率**最高为 **99.213%、99.145%、99.228%、99.460%**，最低为 **73.304%、72.618%、78.760%、74.682%**。

针对子问题 2：本题是订单组批优化问题。首先建立了一个以订单分批效率最高为目标函数的**整数线性规划模型**，通过引入一个不同订单相同材质的相关系数，本文称之为**组批系数**，该模型大大提高了订单组批的效率。在此模型的基础之上，又建立了一个以同批次内板材利用率最高为目标函数的 **0-1 线性规划模型**。进一步实现组批内的板材原片利用率最高。此时，在满足子问题 1 中的所有目标和约束条件的前提下，要求相同的订单下的产品必须在同一批次中，同一批次中的相同材质的产品项可以排样在同一块板材原片上，同时在生产过程中批次的生产的产品数量和产品项面积总和有上限要求。通过建立的模型算法对本题进行求解，计算出了 dataB1~dataB5 的每个数据中订单的组批次分别是：**37、24、23、26、38**。每个数据所使用板材原片数量分别是：**3632、2292、2425、2538、3892**。数据集 B 中 dataB1~dataB5 的板材原片的利用率分别为：**82.217%、81.745%、80.028%、80.297%、84.421%**。

最后，本文所建立的**整数线性规划模型**使用了 **$7n^2+n$** 个决策变量，该模型的复杂度为 **$O(n^2)$**。

关键词：方形件组批优化，全局 0-1 线性规划模型，整数线性规划模型

目 录

一、问题重述

1.1 问题背景

在中国制造 2025 和碳中和、碳达峰的目标背景之下，发展环境保护型、资源节约型的智能制造业已成为制造行业的当务之急。目前企业在智能制造转型的竞争中主要围绕个性化产品定制、系统生命周期、互联互通的服务模式等方面。与此同时，制造业的劳动成本也在不断的升高，生产企业一方面需要降低生产成本，另一方面还要提高生产效率和精细化，在此过程中必然会依赖于机械设备来制造生产各式各样的需求。为了应对客户的个性化定制的拂去需求，企业需要具有高效快速的需求分析及产品设计能力、具有柔性且精益的生产流程、具有完整且精细的全流程生产管控能力。尤其在大批量生产中，即使是对原材料利用率有十分微小的差异，材料的总用量也会有明显的变化。因此想要有效的提高企业的经济效益，关键在于如何降低生产成本，由于原材料费用在生产成本中所占比重较大，因此如何优化下料，并最大限度的提高原材料的利用率是当前很多企业急需解决的重要问题。

根据板材切割工艺的方式可以分为齐头切和非齐头切，齐头切工艺可以细分为精确方式与非精确方式。由切割的阶段数可在三阶段排样方式中分为三种不同的类型：三阶段非精确(3NE)排样方式、三阶段匀质排样方式(3E)、三阶段同质排样方式(3H)。其中 3E 和 3H 排样方式可在三个阶段内切割出准确的方形件，因此属于精确排样方式，而 3NE 排样方式有时需要额外的第四阶段切割才能满足方形件的尺寸要求，因此属于非精准排样方式。

以三阶段切割方式为例，阶段数是指在切割过程中，将板材以垂直于长边或垂直于短边的方向切割生成条带(strip)称之为第一阶段，以垂直于第一阶段切割方向将各个条带切割出多个栈(stack)的方式称为第二阶段，同理以垂直于第二阶段切割方向将各个栈切割出多个产品(item)的方式称为第三阶段。三阶段排样方式示意图如图 1 所示。

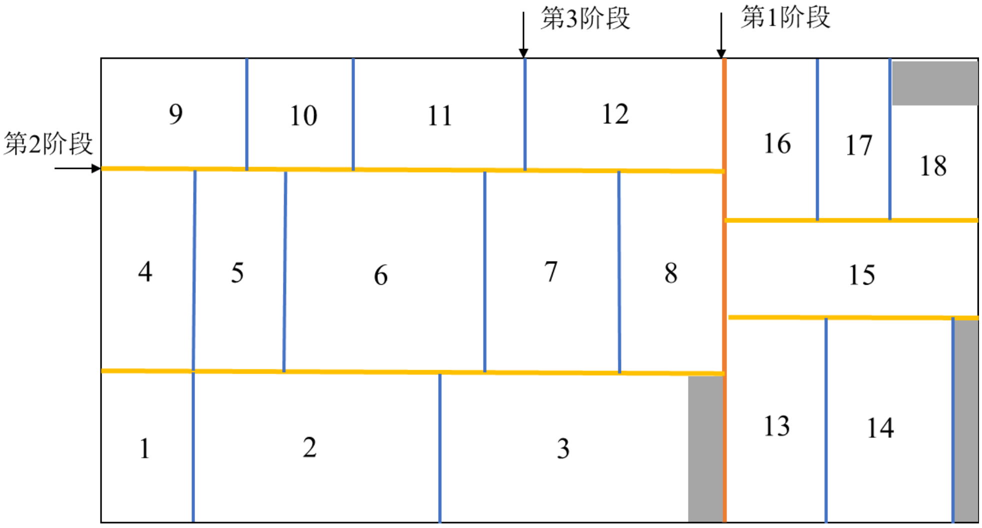


图 1 三阶段排样方式示意图

1.2 问题提出

根据附件提供的信息，包括产品的尺寸、编号、材质和订单号，以及目标要求和约束条件建立完整的数学模型，根据模型设计求解算法，解决以下两个问题：

子问题 1：排样优化问题

根据题目要求建立混合整数规划模型，给不同的产品项进行排版布局，从而实现板材原片使用最少的目标，该问题约束条件为：

1. 在相同栈里的产品项的宽度（或长度）应该相同；

2.最终切割生成的产品项是完整的，非拼接而成。

子问题 2：订单组批问题

要求建立混合整数规划模型，通过建立相应的模型对数据集 B 中所有的订单进行组批，然后对每个批次进行单独排样，在满足子问题 1 的约束条件下，使得板材原片用量尽可能少。本问题在满足问题 1 约束的基础上进一步要求：

1.同一批次中每份订单有且只能出现一次；

2.同一批次中的相同材质的产品项才能使用同一块板材原片进行排样；

3.为保证加工环节快速流转，每个批次产品项总数不能超过 1000；

4.因工厂产能限制，每个批次产品项的面积总和不能超过 250㎡。

二、模型假设与符号说明

2.1 模型假设

假设 1：所有订单的交货期相同

假设 2：只考虑齐头切割方式；

假设 3：订单任务全部完成且不含任何不完整订单任务；

假设 4：板材的切割阶段数不超过 3，处于同一阶段的切割方向一致；

假设 5：排样方式为精确排样，即为均质排样方式（3E）和同质排样方式（3H）；

假设 6：板材原片仅有一种规格且数量充足；

假设 7：排样方案不用考虑锯缝宽度影响。

2.2 符号说明

指标：

符号	含义
i	第 i 个产品项
j	第 j 个栈
k	第 k 个条带
l	第 l 个板块
A	板材原片的原始长度
B	板材原片的原始宽度
a_i	第 i 件产品项的原始长度
b_i	第 i 件产品项的原始宽度
A_i	第 i 件产品项进行旋转之后的长度
B_i	第 i 件产品项进行旋转之后的宽度
Q	一个无穷大的数

决策变量：

符号	含义
r_i	第i个产品不旋转90^o时为1，否则为0，$i\in[1,n]$
u_{ji}	第j个栈中有第i个产品时为1，否则为0，$j\in[1,n]$
v_{kj}	第k个条带中有第j个栈时为1，否则为0, $k\in[1,n]$
ξ_{ji}	$\xi_{ji}=r_i u_{ji}\in\{0,1\}$,满足：$\xi_{ji}\le r_i,\xi_{ji}\le u_{ji},r_i+u_{ji}-1\le\xi_{ji}$
y_{kj}	$y_{ji}=r_i v_{ji}\in\{0,1\}$,满足：$\mathrm{y}_{ji}\le r_i,y_{ji}\le v_{ji},r_i+v_{ji}-1\le y_{ji}$
r_k	第k个条带不旋转90^o时为1，否则为0，$k\in[1,n]$
λ_{lk}	第l个板上有第k个条带时为1，否则为0，$k\in[1,n]$
v_{kk}	第k个条带上有第k个条带时为1，否则为0，$k\in[1,n]$
ζ_k	$\zeta_k=r_k\lambda_{lk}\in\{0,1\}$,满足:$\zeta_k\le r_k,\zeta_k\le\lambda_{lk},r_k+\lambda_{lk}-1\le\zeta_k$

三、子问题的 1 分析与求解

3.1 子问题 1 的分析

本题为排样优化问题，最终实现的目的是板材利用率达到最高，即所有排版的产品项的面积之和比上所使用板材原片面积的比值最大。进一步可以理解为所使用的板材数量尽快能的少。因此，本文针对子问题 1 建立以使用的板材数量最少为目标函数。

本题假定了一些限制条件，包括切割方式、切割阶段、排版样式等相关因素。首先，在本题中只需要考虑齐头切割方式，所谓齐头切割可以理解为是“一刀切”，即每次切割的线条都是直线并且切割的方向都是垂直于板材的一条边，最重要的是要保证每一次切割都需要将板材可以分开成两个部分。其次，板材的切割阶段数不能超过 3，并且处于同一阶段的切割方向必须相同。本题还要求排样的方式是精确排样，精确排样主要有两中类型：三阶段匀质排样方式（3E）和三阶段同质排样方式（3H）。其中匀质排样方式是指在一个相同的栈中存在着具有相同边的不同产品项；而同质排样方式是指在相同的栈中的所有产品都是相同尺寸的产品。本题中也不考虑锯缝宽度所带来的误差。

子问题 1 中的附件数据集 A 中包括了 4 个数据表格，分别是 dataA1~dataA4。对数据进行建模分析之前，首先需要将每一文件中的数值抓取出来。由于文件中给出的数据是统一格式，因而直接利用 Matlab 软件读取.csv 文件，对其进行处理，将其中的 item_length 和 item_width 列进行预处理，对这两列进行取最小值（min）和最大值（max）的处理，分别作为 min 列和 max 列对原表进行整理，并将 max 列作为条件进行降序排序，排序的结果形式如表 1 所示：

表 1 原始表格预处理

序号	item_id	item_num	item_length	item_width	min	max
1	235	1	2418	58	58	2418
2	356	1	2418	58	58	2418
3	542	1	2418	58	58	2418
4	723	1	2416	38	38	2416
5	765	1	2398	393	393	2398
6	113	1	2398	328	328	2398
7	25	1	2398	299	299	2398
8	170	1	2393	38	38	2393
9	115	1	2388	558	558	2388
10	73	1	2388	98	98	2388
11	304	1	2368	328	328	2368
12	61	1	2367	218	218	2367
13	212	1	2353	328	328	2353
14	588	1	2353	58	58	2353
15	48	1	2353	38	38	2353
16	8	1	2348	630	630	2348
17	9	1	2348	558	558	2348
18	479	1	2338	588	588	2338
19	758	1	2334.5	311	311	2334.5
...	…	…	…	…	…	…
89	87	1	177	197.5	177	197.5

其次，本文先将表格中所有的产品的面积之和求出来，其中 dataA1，dataA2，dataA3，dataA4 的所有产品的面积之和分别为：248685614.6、246700070.9、249244736.8、243659621.7；之后题目中确定了板材原片的长度 2440mm 和宽度 1220mm，求出板材原片的面积为 2976800。通过所有产品的面积来确定每一文件所需要原片的极限数量，通过所

有产品面积除以板材原片的面积分别求出了每一文件的最节省的板材数量，所使用最省的板材原片的数量分别为 84、83、84、82。通过数据的预处了可以求出理论上的极限最省板材原片的数量。所以在之后的建模求解过程中不断的进行优化，使得最终求解出的板材数量尽可能的接近理论值，如表 2 所示。

表 2 理想情况下板材原片使用数量及利用率

数据表	data A1	data A2	data A3	data A4
所有产品项的总面积	248685614.6	246700070.9	249244736.8	243659621.7
板材原片的面积	250051200	247074400	250051200	244097600
理论最少板材原片所需数量	84	83	84	82
理论板材原片最大利用率	99.45%	99.85%	99.68%	99.82%

通过对附件中的数据进行分析发现，所有的产品项都属于个性定制的类型，每一个产品的数量均是 1 个。由此，首先考虑的是采用三阶段匀质排样方式（3E），分析题目可以得出，本题是下料优化问题，也称排样优化问题。根据同一生产批次内方形件的尺寸与数量，选择原片的规格和数量，进行下料排样优化，得到最大化板材原片利用率。

本文从每一个产品作为分析题目的出发点，首先由多个相同长度或者宽度的产品项（item）组合成一个栈（Stack），之后由多个宽度或长度相同的栈组成一个条带（Stripe）。最后通过拼接条带使其满足长度和宽度均不超过原片的长度和宽度。同时允许在拼接的过程中存在冗余，即可包含以下废料，通过这种思路找到板材面积利用率最大的切割方式，最终求出最优解，子问题 1 的思路框图如图 2。

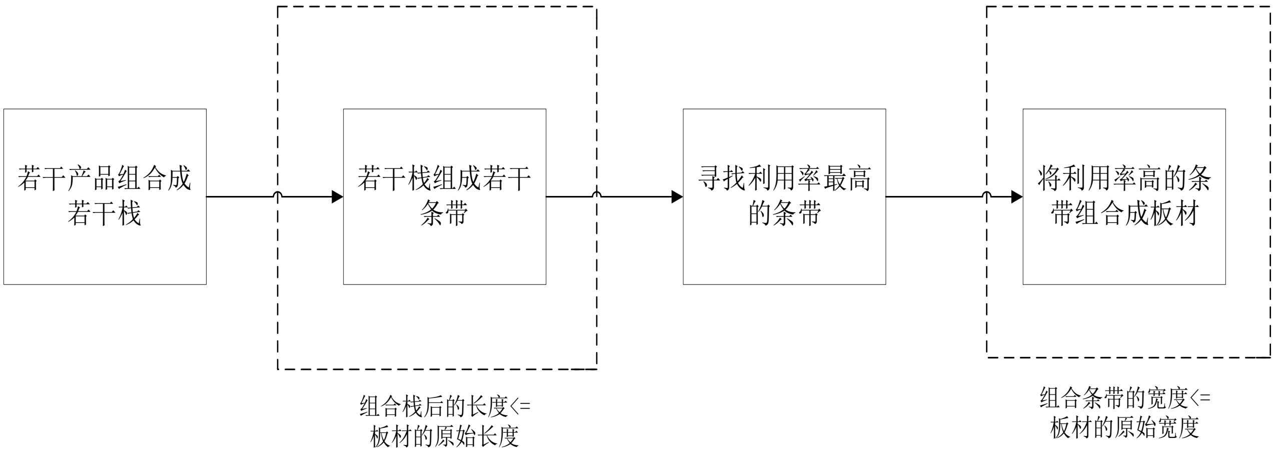


图 2 子问题 1 寻找最优排样方式思路框图

3.2 全局 0-1 线性规划模型的建立

针对子问题建立的全局 0-1 线性规划模型的建立，**本文参考了【文献 3】的带约束条件下的整数线性规划模型。【文献 3】中所建立的模型中并未考虑到产品件可以进行旋转排样，仅仅保持产品件按照固定长宽进行排样。本文重点考虑了产品件可以旋转的情况，在原模型的基础之上进行改进，**原模型结果和改进结果如图 3，4 所示。

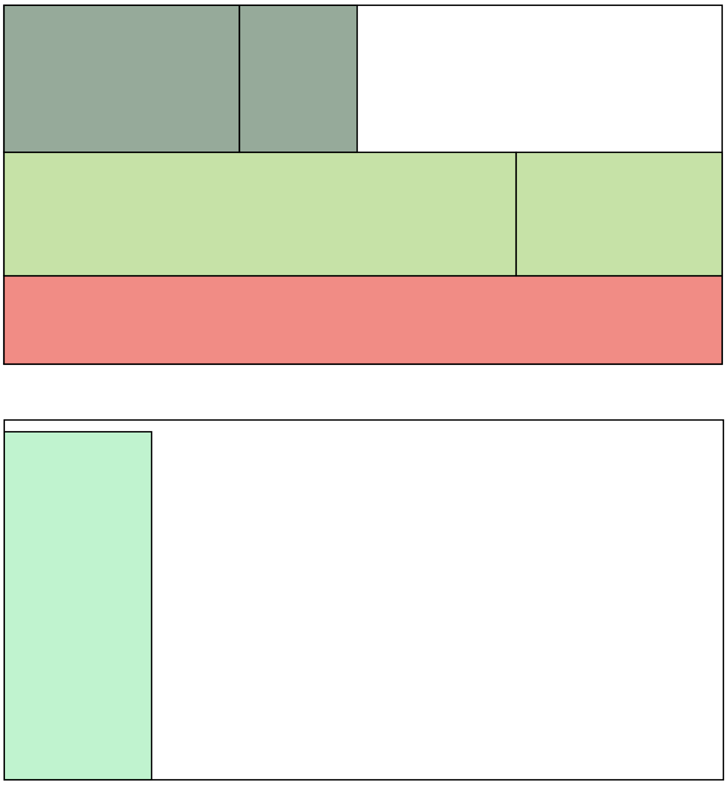


图 3 文献[3]原模型不考虑旋转的排样示意图

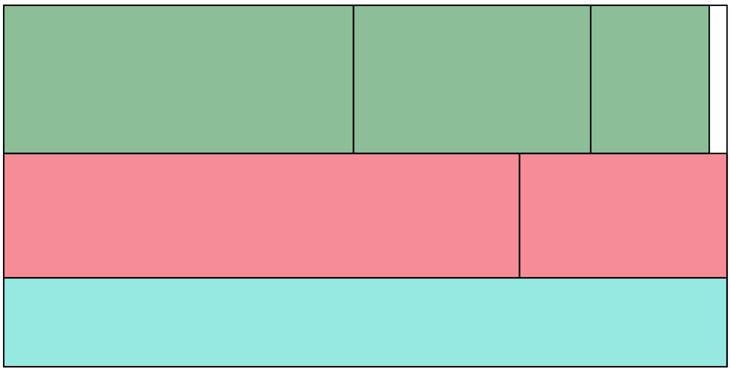


图 4 本文改进后模型考虑产品旋转的排样示意图

通过考虑产品件旋转问题，从图 3，4 可以看出，原始模型需要使用到两块板材，而考虑了产品的旋转之后，只需要一块板材，板材的利用率显著提高。

以下是针对子问题 1 的模型建立过程：

本题为排样优化问题，根据本题的要求首先假设有 i 种规格的产品项需要切割，每块板材原片可以切割出 n 件产品项。用 a_i 来表示第 i 件产品的原始的长度，b_i 来表示第 i 件产品的原始的宽度。题目中所用的板材原片的尺寸都是统一的，用 A 表示板材原片的长度，B 表示板材原片的宽度。

由于在排样的过程中会出现不同的产品可以通过旋转来进行拼接成一个栈，本文考虑了产品件可以旋转的可能性。在此引入了一个决策变量 r_i，公式如下：

$$r_i = \begin{cases} 1, & \text{第i件产品不旋转}90^\circ \\ 0, & \text{第i件产品旋转}90^\circ \end{cases} \tag{1}$$

同时，令产品旋转之后的长度和宽度分别为 A_i 和 B_i，公式如下：

$$\begin{cases} A_i = a_i r_i + b_i(1-r_i) \\ B_i = a_i(1-r_i) + b_i r_i \end{cases} \tag{2}$$

产品项旋转的示意图，如图 5 所示：

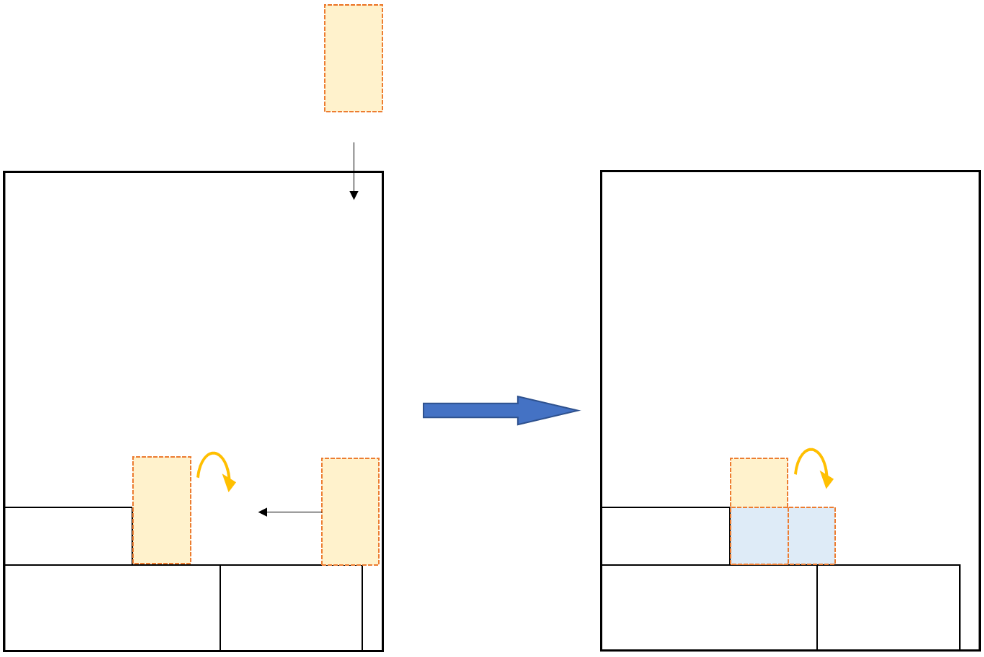


图 5 产品项旋转排样示意图

本文假定在排样的过程中规定第 1 件产品放在板材原片的左下角，并且规定板材原片的左下角为二维坐标的起始原点。在放置的过程中优先将相同的产品项组合成一个栈，并将标号最小的放置在栈中的左下角，同时每个栈的命名均已左下角的产品项的编号命名，所组成的条带也是按照如此的方式进行组合命名。如图 6 所示；

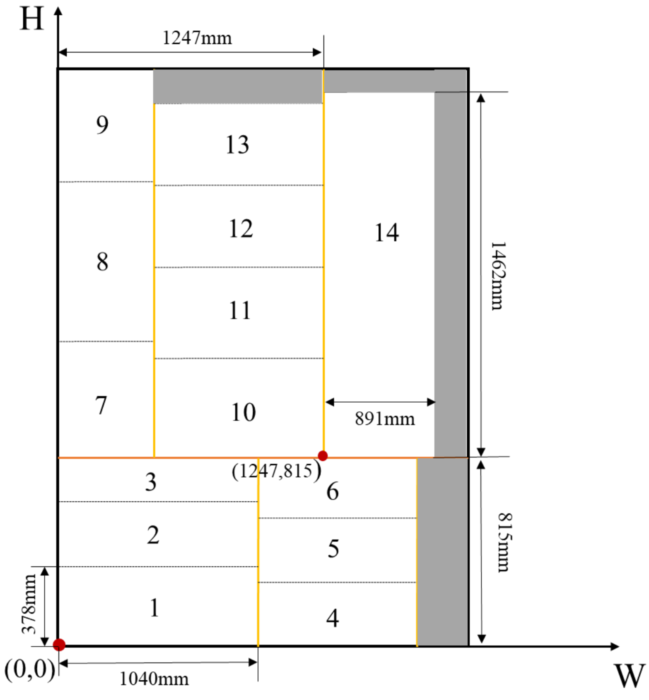


图 6 三阶段精准排样方式示意图

在上述的示意图中，栈的标签分别是 1，4，7，10，14，条带的序号分别是 1 和 7，

同时产品件 1 的左下角为坐标原点(0，0)，由此可以确定产品件 14 的位置为（1247，815）。

根据题目中提供的数据可以得出每件产品的数量都是 1，说明每个栈中所的产品都是唯一确定的。由此引入一个决策变量u_{ji}，表示如下。

$$u_{ji} \in \{0,1\}, j=1,\cdots,n; i=j,\cdots,n$$

u_{ji}代表的是第i件产品在第j个栈上，并且每个栈有且仅有唯一的产品项。当$u_{ji}=1$时表示第i件产品被排在样板上面同时也在对应的j栈上面。当$u_{ji}=0$时，说明不同的产品在组合是长度或者宽度不同或者是所组合之后的长度或宽度超出了板材原片本身的尺寸。由于本文考虑了产品可以旋转的情况，还需要讨论产品旋转的问题，此时每件产品的长度和宽度是变量，分别用A_i和B_i 表示，即公式(3)所示：

$$\begin{cases} A_i = a_i r_i + b_i(1-r_i) \\ B_i = a_i(1-r_i) + b_i r_i \end{cases} \tag{3}$$

当$u_{ji}=0$时需要满足以下条件：

$$B_i \neq B_j \text{或} A_i + A_j > A$$

此时可以从逆否命题进行考虑，即若$u_{ji}=1$时，则：

$$B_i = B_j \text{且} A_i + A_j \leq A$$

通过对表格的数据进行分析发现任意两件产品的长度的差值不超过 2.5mm。由此可以将上式转化为如下形式：

$$\left|B_i - B_j\right| \leq 0.25 \text{且} A_i + A_j \leq A$$

将绝对值展开，此时引入一个无穷大的数Q， 进而推导出：

$$\begin{cases} -2.5-(1-u_{ji})Q \leq B_i - B_j \leq 2.5+(1-u_{ji})Q \\ A_i + A_j \leq A+(1-u_{ji})Q \end{cases} \tag{4}$$

至此，本文引入第二个决策变量v_{kj}，当$v_{kj}=1$时表示第j个栈在第k个条带上面，当$v_{kj}=0$时则表示第j个栈并不在第k个条带上面。由此可以得到一个约束条件：

$$\sum_{k=1}^{j} v_{kj} = u_{jj}, \forall j=1,\cdots,n \tag{5}$$

同时，还需要满足在一个条带中，所包含的栈的长度要大于等于该栈内所有产品的长度之和，其表示如下：

$$\sum_{i=j}^{n} A_i u_{ji} \leq \sum_{k=1}^{j} A_k v_{kj} \tag{6}$$

由于公式中的A_i和A_k是变量，将其用公式展开得：

$$\sum_{i=j}^{n} [a_i r_i + b_i(1-r_i)] u_{ji} \leq \sum_{k=1}^{j} [a_k r_k + b_k(1-r_k)] v_{kj} \tag{7}$$

将其展开之后得：

$$\sum_{i=j}^{n} b_i u_{ji} + \sum_{i=j}^{n} (a_i - b_i) r_i u_{ji} \leq \sum_{k=1}^{j} b_k v_{kj} + \sum_{k=1}^{j} (a_k - b_k) r_k v_{kj} \tag{8}$$

其中，令$\xi_{ji}=r_i u_{ji}\in\{0,1\}$,满足：$\xi_{ji}\le r_i,\xi_{ji}\le u_{ji},r_i+u_{ji}-1\le\xi_{ji}$。
同理，所包含的栈的长度要大于等于该栈内所有产品的长度之和，其表示如下：

$$\sum_{j=k}^{n}B_j v_{kj}\le Bv_{kk} \tag{9}$$

由于公式中的B_j是变量，将其用公式展开得：

$$\sum_{j=k}^{n}[a_j(1-r_j)+b_j r_j]v_{kj}\le Bv_{kk} \tag{10}$$

将其展开之后得：

$$\sum_{j=k}^{n}a_j v_{kj}+\sum_{j=k}^{n}(b_j-a_j)r_j v_{kj}\le Bv_{kk} \tag{11}$$

其中，令$\eta_{kj}=r_j v_{kj}\in\{0,1\}$,满足：$\eta_{kj}\le r_j,\eta_{kj}\le v_{kj},r_j+v_{kj}-1\le\eta_{kj}$。
最后，引入一个决策变量λ_{lk}，当$\lambda_{lk}=1$时表示第k个条带在第l块板材原片上，当$\lambda_{lk}=0$时表示第k个条带不在第l块板材原片上，由此可以得到一个约束条件：

$$\sum_{l=1}^{k}\lambda_{lk}=v_{kk},\forall k=1,\cdots,n \tag{12}$$

与此同时，还需要满足在同一个板材原片上得所有条带得宽度总和要小于等于板材原片的宽度，即：

$$\sum_{k=l}^{n}A_k\lambda_{lk}\le A\lambda_{ll} \tag{13}$$

由于公式中的A_k是变量，将其用公式展开得：

$$\sum_{k=l}^{n}[a_k r_k+b_k(1-r_k)]\lambda_{lk}\le A\lambda_{ll} \tag{14}$$

将其展开之后得：

$$\sum_{k=l}^{n}b_k\lambda_{lk}+\sum_{k=l}^{n}(a_k-b_k)r_k\lambda_{lk}\le A\lambda_{ll} \tag{15}$$

其中，令$\zeta_k=r_k\lambda_{lk}\in\{0,1\}$,满足:$\zeta_k\le r_k,\zeta_k\le\lambda_{lk},r_k+\lambda_{lk}-1\le\zeta_k$，
综上所述，针对子问题 1 建立了一个全局 0-1 线性规划模型，模型如下：

$$\min(\lambda_{11}+\lambda_{22}+\lambda_{33}+\ldots+\lambda_{nn})$$

$$
s.t.\begin{cases}
\sum_{j=1}^{i} u_{ji}=1, \forall_{i}=1,\cdots,n \\
-0.25-(1-u_{ji})Q \le B_i-B_j \le 0.25+(1-u_{ji})Q \\
\sum_{k=1}^{j} v_{kj}=u_{jj}, \forall j=1,\cdots,n \\
\sum_{i=j}^{n} b_i u_{ji}+\sum_{i=j}^{n}(a_i-b_i)\xi_{ji} \le \sum_{k=1}^{j} B_k V_{kj}+\sum_{k=1}^{j}(a_k-b_k)\eta_{kj}, \forall j=1,\cdots,n-1 \\
\sum_{j=k}^{n} a_k v_{kj}+\sum_{j=k}^{n}(b_k-a_k)\eta_{kj} \le B v_{kk}, \forall k=1,\cdots,n-1 \\
\sum_{l=1}^{k} \lambda_{lk}=v_{kk}, \forall k=1,\cdots,n \\
\sum_{k=l}^{n} b_k \lambda_{lk}+\sum_{k=l}^{n}(a_k-b_k)\zeta_{lk} \le A\lambda_{ll}, \forall l=1,\cdots,n-1 \\
u_{ji}\in\{0,1\}, j=1,\ldots,n, i=j,\ldots,n \\
v_{kj}\in\{0,1\}, k=1,\ldots,n, j=k,\ldots,n \\
\lambda_{lk}\in\{0,1\}, l=1,\ldots,n, k=l,\ldots,n \\
\xi_{ji}\in\{0,1\}, j=1,\ldots,n, i=j,\ldots,n \\
\eta_{kj}\in\{0,1\}, k=1,\ldots,n, j=k,\ldots,n \\
\zeta_{lk}\in\{0,1\}, l=1,\ldots,n, k=l,\ldots,n \\
r_i\in\{0,1\}, i=1,\ldots,n
\end{cases}
$$

3.3 子问题 1 的算法分析与结果

首先对**子问题 1 的算法复杂度进行分析，**参考文献【3】中原始的整数线性规划模型中的使用了 $3n^2$ 个决策变量，即模型的时间复杂度为 $O(n^2)$，约束条件为 O(n)。本文**改进后的模型**使用了 $7n^2+n$ 个决策变量，也为 $O(n^2)$，约束条件为 O(n)。

其次，针对本题所要求的“一刀切”的切割约束下的、多个规格类似的产品排样优化问题，采用启发式算进行求解，为了满足本题约束，利用传统的分层排样方式进行排样。该算法思想可以很好的满足“一刀切”的切割约束，具体算法步骤如下：

步骤 1：首先对产品数据进行预处理，先筛选出长和宽最小值为一列，其次筛选出最大为一列，并将所有待排的产品按照长宽最大值优先方式进行从大到小的排序；排序好的数据如表 3 所示：

表 3 原始表格预处理

序号	item_id	item_num	item_length	item_width	min	max
1	235	1	2418	58	58	2418
2	356	1	2418	58	58	2418
3	542	1	2418	58	58	2418
4	723	1	2416	38	38	2416
5	765	1	2398	393	393	2398
6	113	1	2398	328	328	2398
7	25	1	2398	299	299	2398
8	170	1	2393	38	38	2393
9	115	1	2388	558	558	2388
10	73	1	2388	98	98	2388
11	304	1	2368	328	328	2368
12	61	1	2367	218	218	2367
13	212	1	2353	328	328	2353
14	588	1	2353	58	58	2353
15	48	1	2353	38	38	2353
16	8	1	2348	630	630	2348
17	9	1	2348	558	558	2348
18	479	1	2338	588	588	2338
19	758	1	2334.5	311	311	2334.5
...	…	…	…	…	…	…
89	87	1	177	197.5	177	197.5

步骤 2：其次将产品进行横向逐层排样，当排样到下一个产品时，若当前层剩余的空间不足,则选择下一个合适的产品进行排样。

步骤 3：如果在当前层排样下一个产品的时候，没有一个符合要求的产品，则以该层中宽度最大的产品作为最高基准线进行向上新一层的排样。

步骤 4：重复上述（2）（3）步骤，直到排完所有产品。

传统的分层排样方式是利用最高的产品作为水平基准线将板材分为多层。该算法的排样方式如图 7 所示：

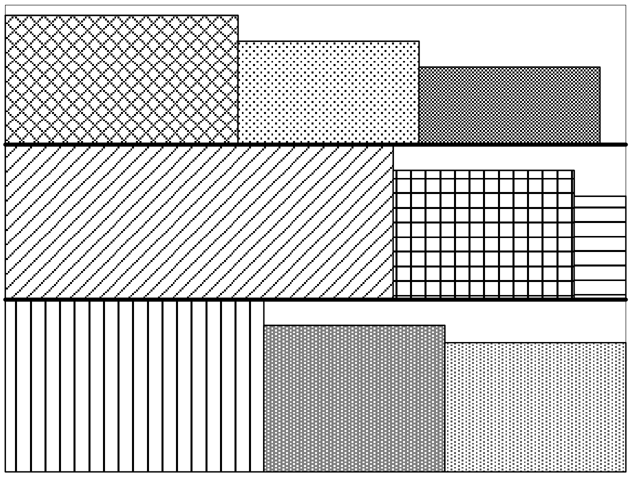


图 7 传统排样方式

传统算法进行改进后的最低基准线搜索算法具体步骤如下：

先对产品数据进行预处理，先筛选出长和宽最小值为一列，其次筛选出最大为一列，并将所有待排的产品按照长宽最大值优先方式进行从大到小的排序；

首先将第一个产品放在板材的左下角，并与边界对齐，并形成水平基准线，其首末位置代表基准线宽度范围。排样产品的时候搜索高度最低的基准线进行排样，有以下三种情况需要处理。

情况 1：如果只有一条最低的水平基准线，则利用最佳匹配搜索算法在剩余的产品中找到最适合的，将其与待排样产品调换。在排样的过程中必须满足在此产品后的排样高度不能超过板材高度，若超出，则在剩余的产品中搜索满足要求的产品进行排样，若没有，则将剩余的产品在新的板材上排样。排样后更新水平基准线，更新后的基准线首末位置不能超过该层所对应的宽度范围。该排样方式如图 8 所示。

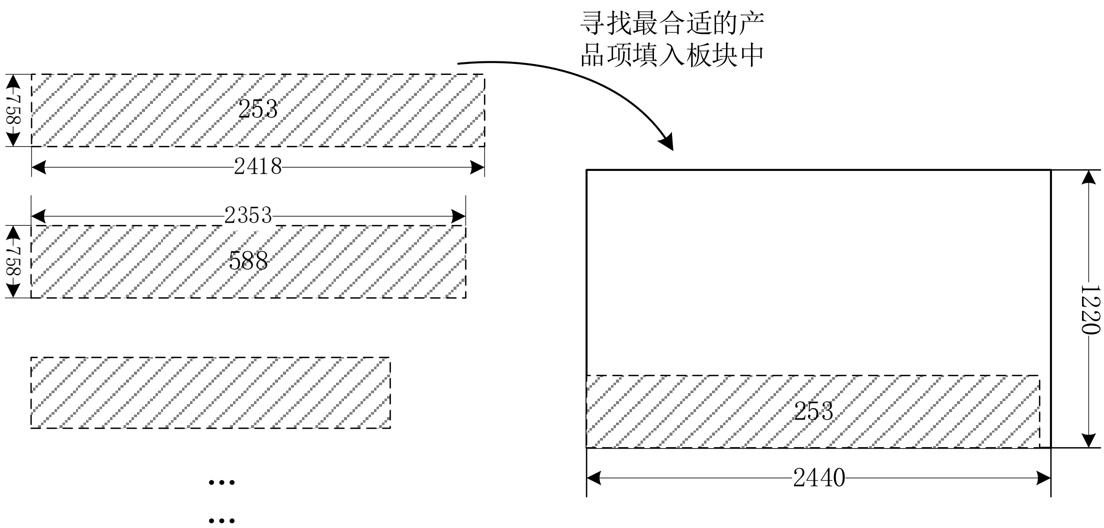


图 8 寻找最适合产品项

情况 2：如果有多条最低水平基准线，则搜索长度最适合的产品排样在左下角，排样后更新水平基准线，同样考虑 1）中的板材高度和基准线宽度的限制。该步骤排样方式如

图 9 所示。

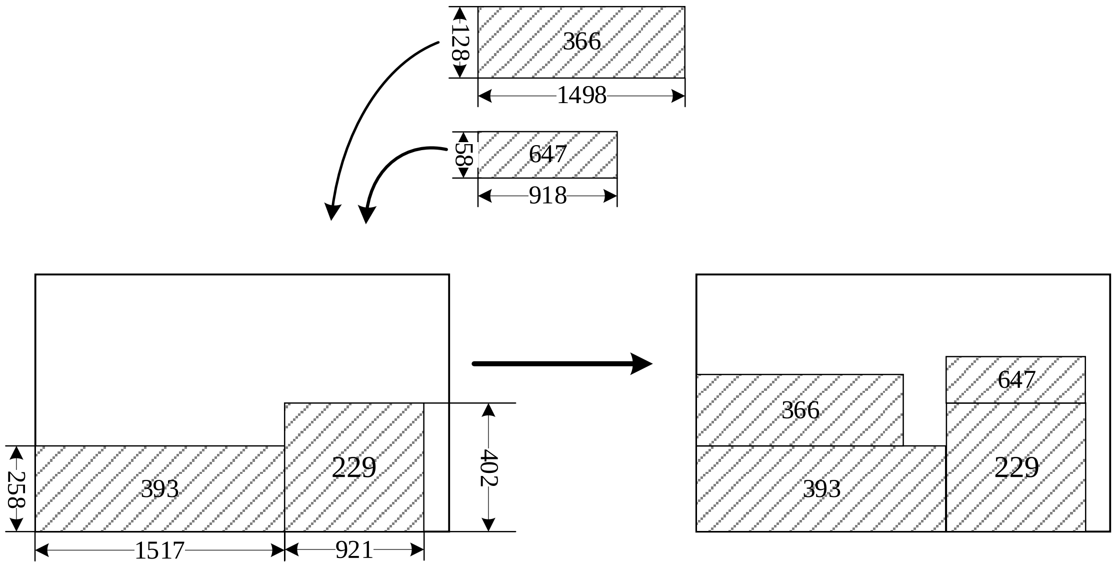


图 9 多条基准线产品排样方式

情况 3：如果所有的最低水平基准线都不能放下该产品，则继续搜索剩余待排样的产品并更新水平基准线；如果最低水平基准线长度小于所有剩余产品的长度，则将其旋转 90°后再寻找长度合适的产品，将其与之前待排样的产品调换位置，如果该产品在旋转 90°后仍然不能在最低基准线排样，则将该产品移放在相邻基准线的位置，排样后更新水平基准线，同样考虑 1 中的板材高度和基准线宽度的限制。该步骤排样方式如图 10 所示。

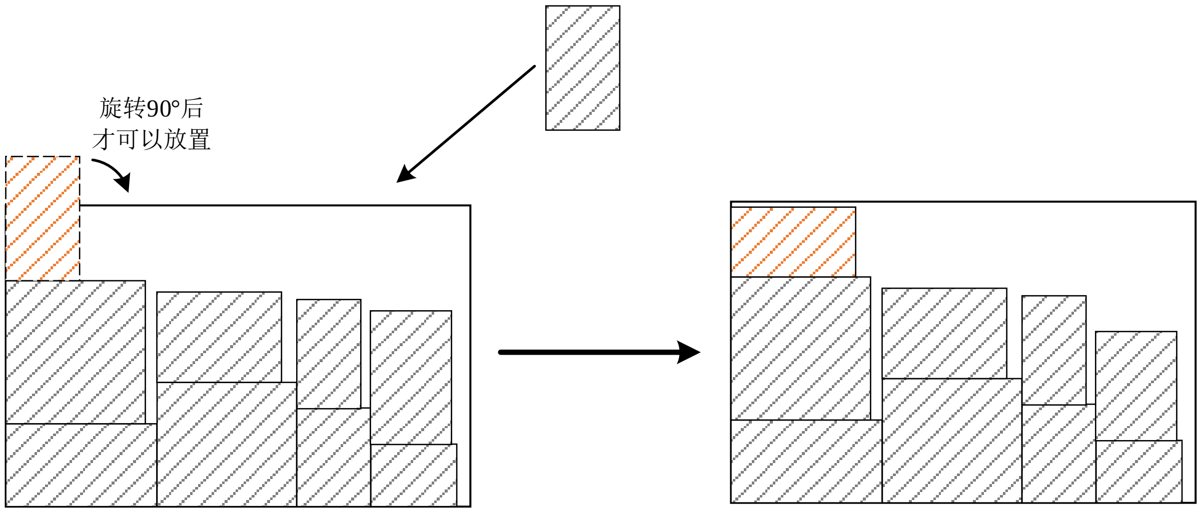


图 10 产品旋转排样方式

重复上述步骤，直到排样完所有产品。改进后的算法排样方式如图 11 所示：

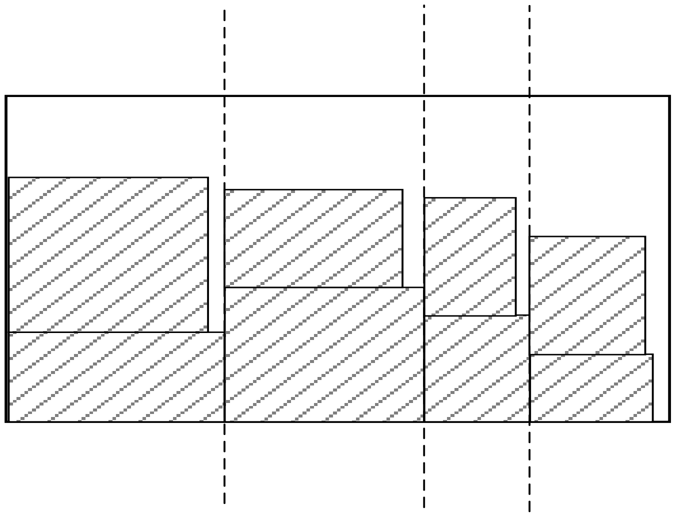


图 11 改进算法后的产品排样方式

在实际的算法实现中我们还可以继续对该算法进行优化，在保证板材利用率最高的情况下，我们可以将传统的分层排样方式与改进后的最低基准线搜索算法相结合，在处理实际产品排样的时候，我们可以定义一个条件判断语句，将两种算法同时使用，取其中利用率最高的一种方式，得出该板材最高的利用率。算法主要流程如图 12 所示：

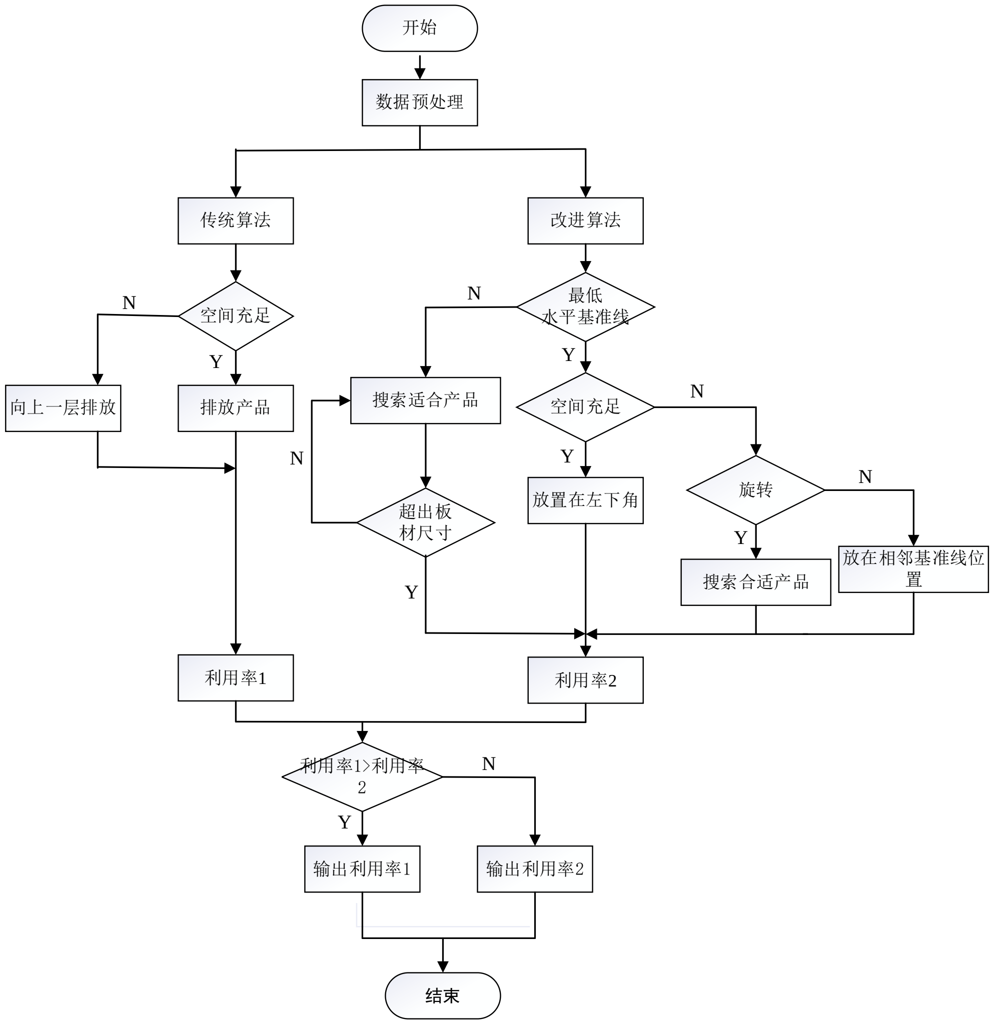


图 12 算法流程图

下面从附件中 dataA1 数据中具体阐述算法具体的实现步骤。

步骤 1：采用启发式算法，利用 dataA1.csv 所提供的数据作为参考数据，首先考虑尺寸最大的产品项被切割出来。利用搜索的办法，找到长度最长的产品项。可以发现有三个数据，分别为 231、347、527 组数据，长、宽分别为 2418、58，另外一个数据随即被搜索出来，即长度第二的一个数据为第 703 组，长、宽分别为 2416、38。

步骤 2：首先，第一刀切割，将一块板材分为两个条带。一个条带的宽为 $58\times3+38=212$，另一条带的宽则为 $1220-212=1008$。此时，仍然使用启发式搜索的办法，找到长度恰好为 1008 的产品项，其宽度为 218，考虑到最大利用率，我们可以利用该工艺切割出该产品项，余下的条带可以继续切割。考虑到使用 3E 的方式切割，因为数据表中给出的产品项都是个性化定制且订货量只有一种，即便几种产品项的尺寸可能相同，搜索不到两种产品项的组合使得长度和为 1008 或 $2440-298=2142$ 的。

步骤 3：接着尝试使用三种产品项的组合，条件为宽度和为 1008，长度相同，找到两组数据，其中一组为 351,368,621 组产品项，长度为 1998，宽度分别为 580、330、98。再余下的条带长为 144，继续搜索得到宽在(132，144)之间的产品项。经观察，可得到第 244，424 组数据的长宽分别为 655 和 132.5 和 353 和 132.5。

步骤 4：最后根据利用率公式：

板材利用率 = 产品项面积之和 / 使用板材原片面积之和

通过上式可以计算出，此时的板材的利用率为 99.45%。如图 13 所示。

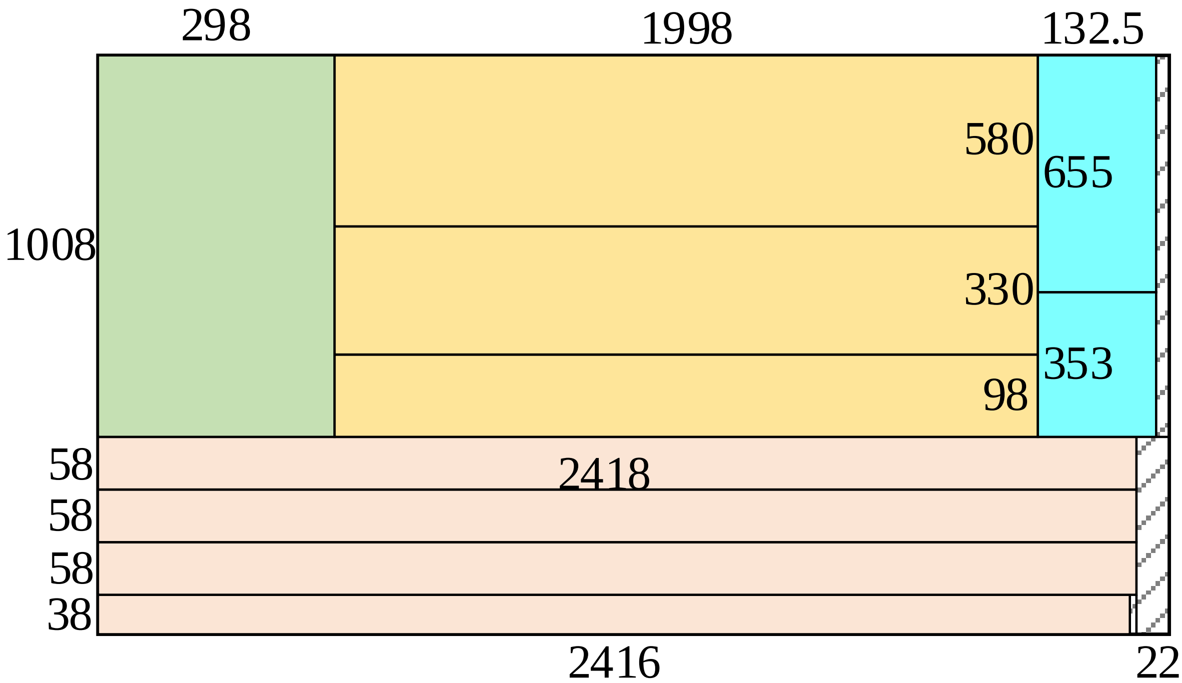


图 13 某一板材排版方式

通过建立的子问题 1 的模型算法，求解出了 dataA1~dataA4 中的实际板材使用的数量和实际使用效率。如图 14 所示。

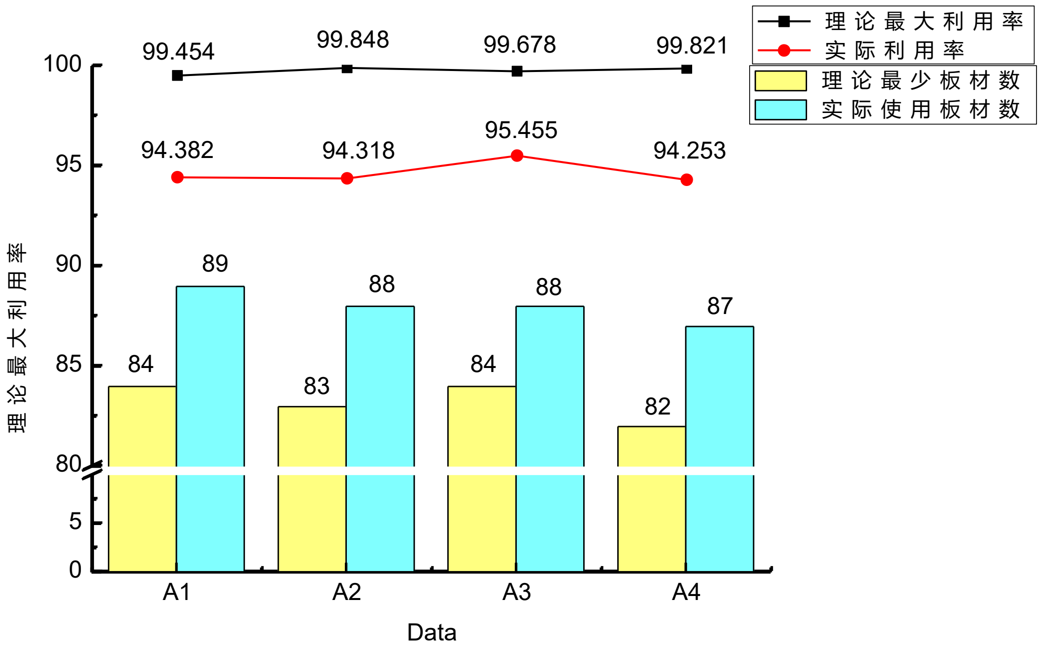


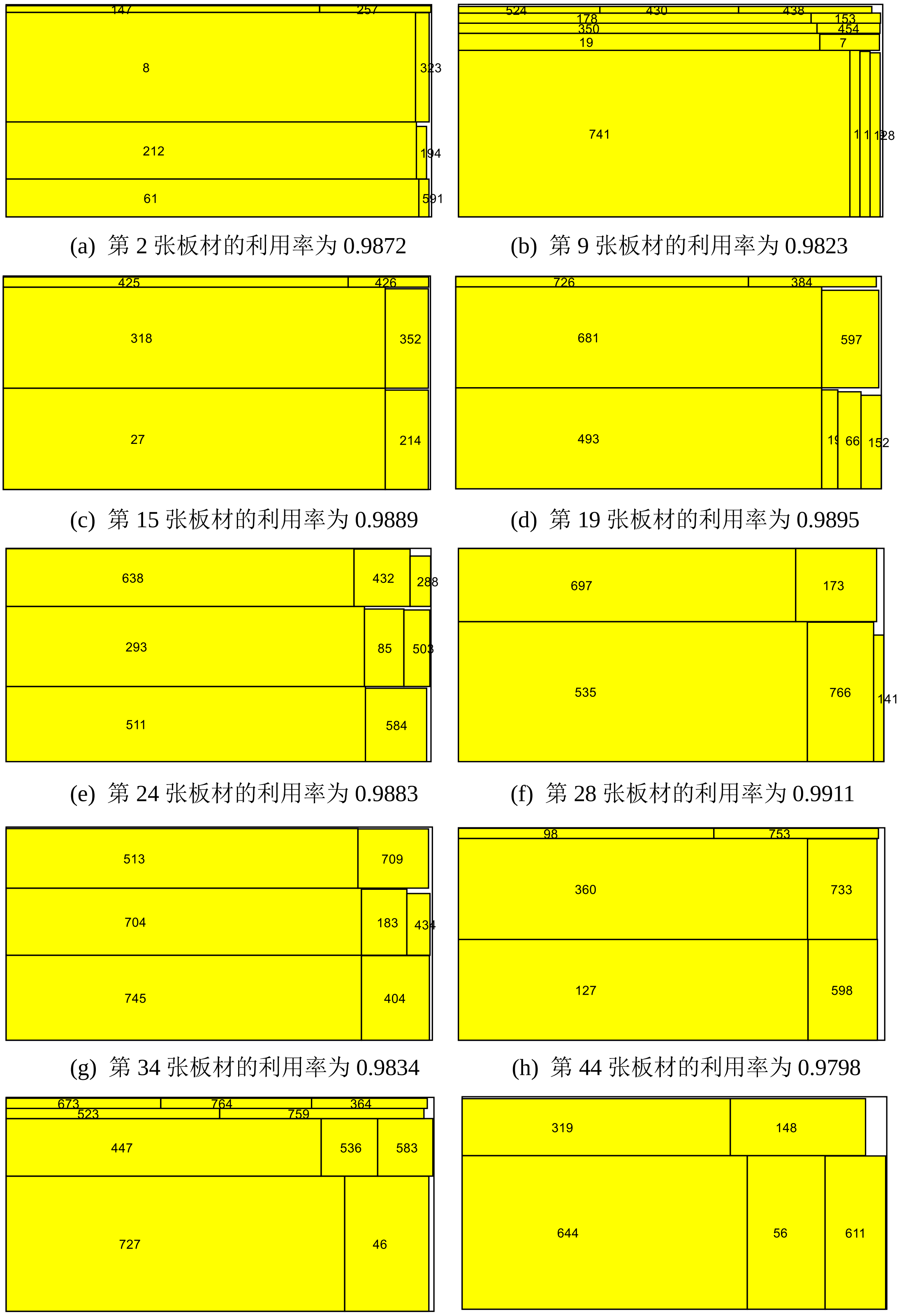
图 14 dataA1~dataA4 中板材原片的使用数量和利用率

根据子问题要求的排版方案，以板材原片的左下角作为坐标原点，通过二维坐标系的形式来描述每个产品项的位置分布，最终的输出结果以 dataA1 为例如表 4 所示，其他数据在附件中。

表 4 dataA1 排版的输出结果

原片材质	原片序号	产品 id	产品 x 坐标	产品 y 坐标	产品 x 方向长度	产品 y 方向长度
YW10-0218S	0	235	0	0	2418	58
YW10-0218S	0	356	0	58	2418	58
YW10-0218S	0	542	0	116	2418	58
YW10-0218S	0	723	0	174	2416	38
YW10-0218S	0	765	0	212	2398	393
YW10-0218S	0	286	2398	212	38	356
YW10-0218S	0	113	0	605	2398	328
YW10-0218S	0	309	2398	605	38	314
YW10-0218S	0	170	0	933	2393	38
…	…	…	…	…	…	…
YW10-0218S	89	354	0	0	444	443

以 dataA1 数据为例，随机选取了其中 10 张排样方式图，结果如图 15 所示。



(a) 第 2 张板材的利用率为 0.9872　(b) 第 9 张板材的利用率为 0.9823

(c) 第 15 张板材的利用率为 0.9889　(d) 第 19 张板材的利用率为 0.9895

(e) 第 24 张板材的利用率为 0.9883　(f) 第 28 张板材的利用率为 0.9911

(g) 第 34 张板材的利用率为 0.9834　(h) 第 44 张板材的利用率为 0.9798

(i) 第 57 张板材的利用率为 0.9859　(j) 第 58 张板材的利用率为 0.9754

图 15 dataA1 随机 10 张产品排样方式图

根据子问题 1 中的要求，附件中 A 数据集中的板材原片的利用率，如表 5 所示。

表 5 dataA1~dataA4 中的板材原片利用率

结果指标	dataA1	dataA2	dataA3	dataA4
板材利用率	94.382%	94.318%	95.455%	94.253%

四、子问题的 2 分析与求解

4.1 子问题 2 的分析

本题为订单组批问题，需要建立相应的混合整数规划模型，对附件中的数据中全部的订单进行组批，同时还需要对每一个批次进行单独排版，在满足子问题 1 的所有条件的基础之上，最终实现板材原片的使用数量尽可能的少。

本题是建立在子问题 1 中的所有约束条件的基础上进一步进行约束，既要满足同一栈中所有的产品项的宽度或长度应该相同，每件产品所必须完整的切割出来，不可以进行拼接，还要满足只有相同材质的产品项才可以放在同一原片上进行排样，并且订单相同的产品有且仅有放在同一个批次之内，同时每个批次之内的产品项的数量不可以超过 1000 个，每一批次的总面积也不可超高 250m^2。

子问题 2 中的附件中包括了 5 个数据表格，分别是 dataB1~dataB5。在对本题建模之前需要对原始的数据进行数据预处理，先按照相同的订单号（item_order）对数据进行排序，其次对同一订单之内的产品的材质（item_material）进行排序。通过对数据处理，可以清晰的看出每个数据中所包含的订单数量和板材的材质数量，如表 6 所示：

表 6 子问题 2 附件中订单数据和板材数据

	Data B1	Data B2	Data B3	Data B4	Data B5
订单数	545	402	409	380	603
板材数	130	146	160	142	192

通过建立整数线性规划模型，最终求解出在本题的约束条件之下，板材原片的使用数量。本题的分析思路如图 16 所示。

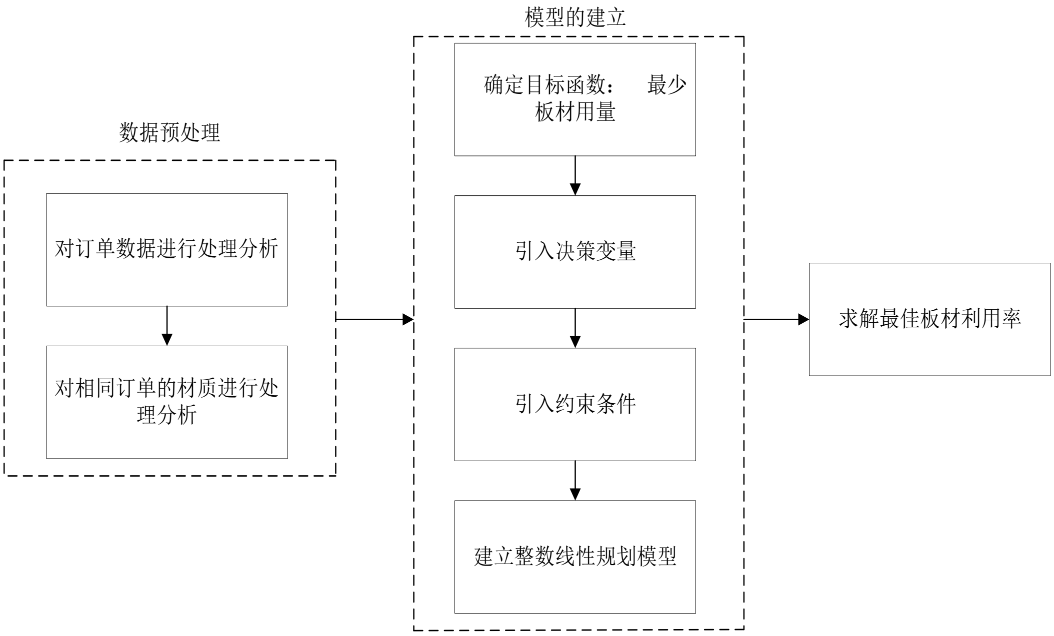


图 16 子问题 2 的分析思路图

4.2 订单组批模型的建立

本题为订单组批问题，根据本题的要求首先对子问题 2 附件中的数据进行预处理，得到每个数据表格中的所有的产品的订单数量和所有板材原片的材质数量。以附件 dataB1 为例，dataB1 中包含了 546 个订单号，130 种板材数量。本题所建立的整数线性规划模型，其约束条件是在满足子问题 1 的所有约束条件基础之上，还需要满足相同订单在同一批次中、产品项的材质要与板材原片的材质保持一致。

首先记订单 d 内有 n_d 件产品项，$d=1,...,o$，其中材质为 e 的产品项有 n_{de}，此时在订单 d 中的所有材质为 e 的产品项为：

$$\sum_{e=1}^{M} n_{de} = n_d$$

由于题目中限制了单个批次产品项的总数不得超过 1000，在订单组批的过程中出现了产品项的数量远远不足 1000，此时为了尽量是每个批次的产品项的数量尽可能的多，并切每个批次的产品尽可能的均衡。此时引入一个相关系统并命名为组批系统，记为 θ。对于订单 d_1 中材质为 e 的产品的长度或宽度与订单 d_2 中材质为 e 的产品的长度或者宽度可以组合拼接的组数比上在订单 d_1 中材质为 e 的数量与订单为 d_2 中材质为 e 的数量乘积，可以表示如下：

$$\theta_{d_1 d_2 e} = \frac{\sum_{e=1}^{M} N_{d_1 d_2 e}}{n_{d_1 e} n_{d_2 e}}$$

其中 $N_{d_1 d_2 e}$ 表示为在 d_1 和 d_2 订单中材质均为 e 并且可以拼接的产品项，进一步可化为：

$$\theta_{d_1 d_2 e} = \frac{1}{n_{d_1 e} n_{d_2 e}} \sum_{k=1}^{n_{d_1 e}} \sum_{l=1}^{n_{d_2 e}} \psi_{kl}^{(d_1 d_2 e)}$$

其中，$\psi_{kl}^{(d_1 d_2 e)}$ 是引入的决策变量，表示如下：

$$\psi_{kl}^{(d_1 d_2 e)} = \begin{cases} 1, \text{在订单} d_1\text{，} d_2 \text{中材质相同且可以组合成栈的产品} \\ 0, \text{否} \end{cases}$$

令 $\theta_{d_1 d_2} = \sum_{e=1}^{M} \theta_{d_1 d_2 e}$，将其称为 d_1 与 d_2 的总组批系数。表 x 是对 dataB1 中的数据进行处理，得出不同订单中相同材质的组批系数 θ。此时，引入决策变量 $\sigma_{d_1 d_2}$，当 $\sigma_{d_1 d_2} = 1$ 时表示订单 d_1 和 d_2 被安排同一个批次中，即公式所示：

$$\sigma_{d_1 d_2} = \begin{cases} 1, \text{订单} d1 \text{与} d2 \text{被安排在同一个批次内} \\ 0, \text{否} \end{cases}$$

此时，针对于子问题 2 的订单组批模型的目标是在同一批次内不同订单之间的组批系统最大，即：

$$\max \sum_{d_1,d_2} \theta_{d_1 d_2} \sigma_{d_1 d_2}$$

综上所示，针对于子问题 2 建立了一个整数线性规划模型，模型如下：

$$\max \sum_{d_1,d_2} \theta_{d_1 d_2} \sigma_{d_1 d_2}$$

$$s.t.\begin{cases} \sum_{d_2=1}^{o} n_{d_2} \sigma_{d_1 d_2} \le 1000, \forall d_1 = 1,...,o \\ \sum_{d_2=1}^{o} a_{d_2} b_{d_2} \sigma_{d_1 d_2} \le 250, \forall d_1 = 1,...,o \\ \sigma_{d_1 d_1} \equiv 1 \\ \sigma_{d_1 d_2} \equiv \sigma_{d_2 d_1} \\ \sigma_{d_1 d_2} \in \{0,1\}, \forall d_1, d_2 = 1,...,o \end{cases}$$

4.3 同批次排样优化模型的建立

在上述所建立的线性模型基础之上，还需要对每块板材原片的排样进行优化，建立出相应的整数线性规划模型，使在同一批次之内所使用的板材的利用率最高。此时的排样方式优化模型与子问题 1 的模型相似，推导公式与过程与子问题 1 相同，在子问题的模型之上，考虑了板材的批次和板材的材质问题，故可以建立一个整数线性规划模型，模型如下。

$$\min \sum_{d1,d2} (\lambda_{11} + \lambda_{22} + \lambda_{22} + ... + \lambda_{22}) \sigma_{d_1 d_2}$$

$$s.t.\begin{cases} \sum_{j=1}^{i} u_{ji} = 1, \forall_i = 1,\cdots,n \\ -0.25 - (1-u_{ji})Q \le B_i - B_j \le 0.25 + (1-u_{ji})Q \\ \sum_{k=1}^{j} v_{kj} = u_{jj}, \forall j = 1,\cdots,n \\ \sum_{i=j}^{n} b_i u_{ji} + \sum_{i=j}^{n} (a_i - b_i)\xi_{ji} \le \sum_{k=1}^{j} B_k V_{kj} + \sum_{k=1}^{j} (a_k - b_k)\eta_{kj}, \forall j = 1,\cdots,n-1 \\ \sum_{j=k}^{n} a_k v_{kj} + \sum_{j=k}^{n} (b_k - a_k)\eta_{kj} \le B v_{kk}, \forall k = 1,\cdots,n-1 \\ \sum_{l=1}^{k} \lambda_{lk} = v_{kk}, \forall k = 1,\cdots,n \\ \sum_{k=l}^{n} b_k \lambda_{lk} + \sum_{k=l}^{n} (a_k - b_k)\zeta_{lk} \le A\lambda_{ll}, \forall l = 1,\cdots,n-1 \\ u_{ji} \in \{0,1\}, j = 1,...,n, i = j,...,n \\ v_{kj} \in \{0,1\}, k = 1,...,n, j = k,...,n \\ \lambda_{lk} \in \{0,1\}, l = 1,...,n, k = l,...,n \\ \xi_{ji} \in \{0,1\}, j = 1,...,n, i = j,...,n \\ \eta_{kj} \in \{0,1\}, k = 1,...,n, j = k,...,n \\ \zeta_{lk} \in \{0,1\}, l = 1,...,n, k = l,...,n \\ r_i \in \{0,1\}, i = 1,...,n \end{cases}$$

4.4 子问题 2 的算法分析与结果

本题的附件中给出了数据 B，其中包含了五个表格，分别是 dataB1~dataB5。本题在输入数据说明中明确描述了关于板材原片的材质、产品项的数量、产品项的材质、产品项的尺寸和订单号等各项信息。本文在编写代码时，首先对数据进行预处理，详细的算法步骤如下：

第一步：将输入的各项参数进行定义与赋值

第二步：读取 dataB 数据中的产品件的订单号(item_order)数据，并对订单号数据进行处理将相同的订单号进行聚类和排序，并且设定产品件安排在同一批次的约束条件。

第三步：读取 dataB 数据中产品材质(item_material)列，将在同一批次内的产品项按照相同材质进行分析处理。

第四步：通过子问题 1 中的排样方式模型，对子问题 2 中的处于同一批次内的产品，同时还需满足材质相同的产品进行排样优化，最后实现最优的板材利用率。

第五步：根据上述步骤建立相应的整数线性规划模型，编写代码。

本文将 B 组数据中按照所建立的整数线性模型算法进行处理，数据集 B 中板材原片的订单组批如表 7，利用率如表 8 所示。

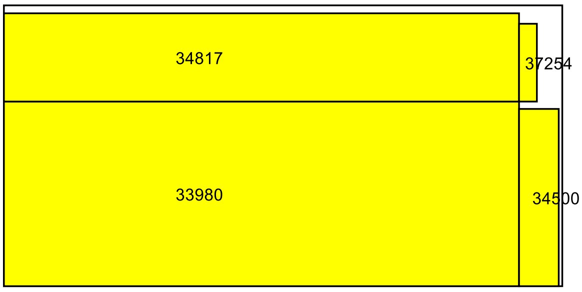
表 7 dataB1~dataB4 中的订单组批数量

结果指标	dataB1	dataB2	dataB3	DataB4	DataB5
订单组批数	37	24	23	26	38

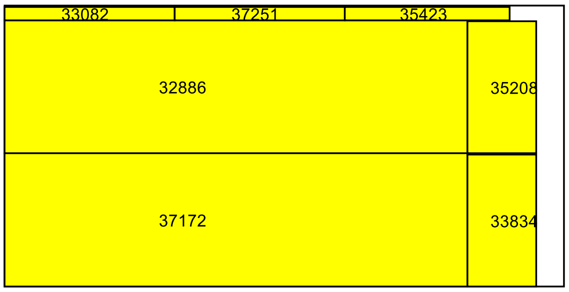
表 8 dataB1~dataB4 中的板材原片利用率

结果指标	dataB1	dataB2	dataB3	DataB4	DataB5
板材利用率	82.217%	81.745%	80.028%	80.297%	84.421%

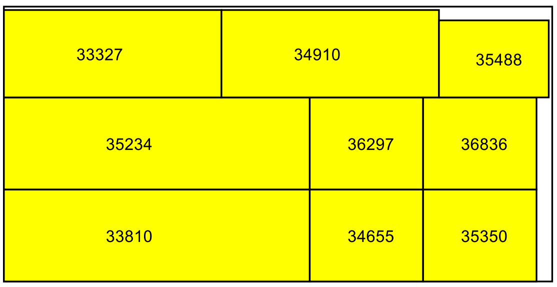
根据子问题 2 所建立的模型，随机从 dataB1~dataB5 中每个数据随机选 4 个排版方式，如图 17 所示。其余排板方式见附件。



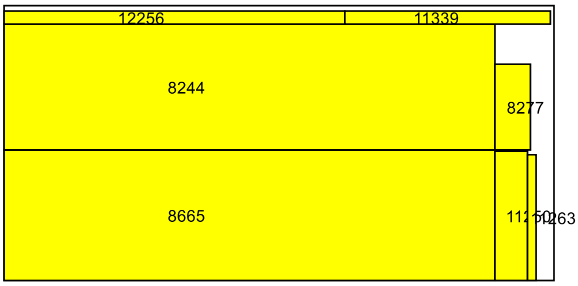
(a) B1 第 1 张板材(JH-02185)利用率为 0.9501



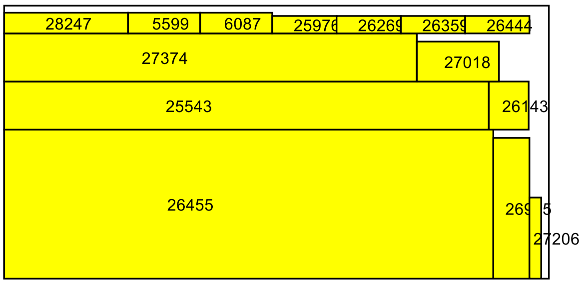
(b) B1 第 32 张板材(JH-02185)利用率为 0.9425



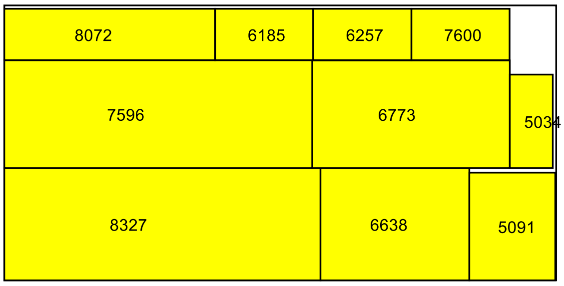
(c) B1 第 85 张板材(JH-02185)利用率为 0.9589



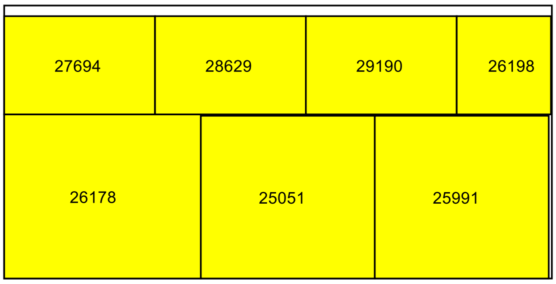
(d) B1 第 143 张板材(JH-02185)利用率为 0.9350



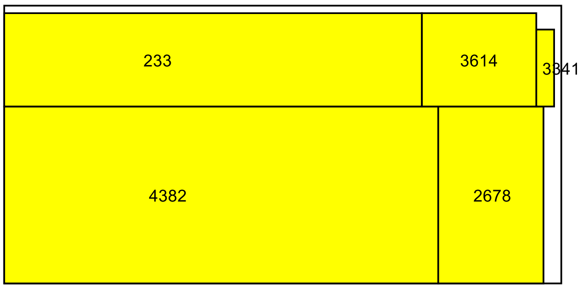
(e) B2 第 927 张板材(HYBW-02185)利用率为 0.9236



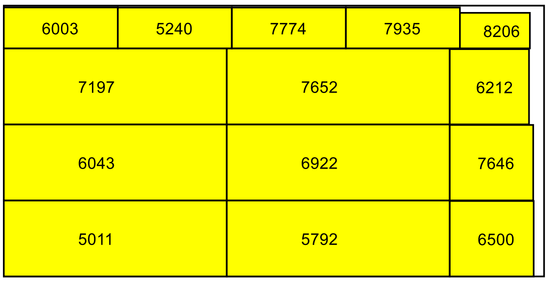
(f) B2 第 588 张板材(5-02185)利用率为 0.9501



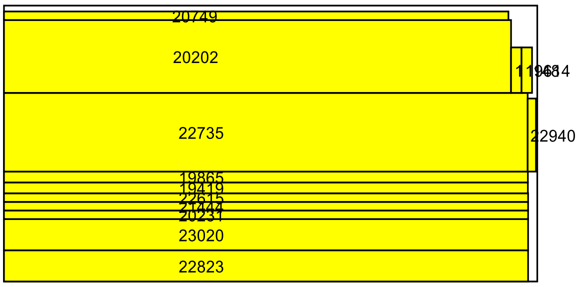
(g) B2 第 401 张板材(FBM-0215S)利用率为 0.9539



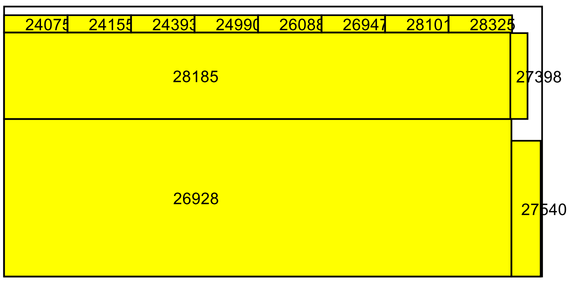
(h) B2 第 10 张板材(JH-0215S)利用率为 0.9464



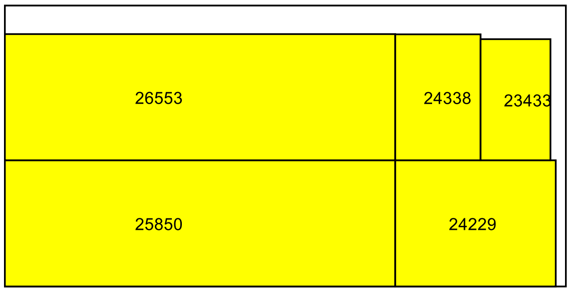
(i) B3 第 778 张板材(5-0215S)利用率为 0.9667



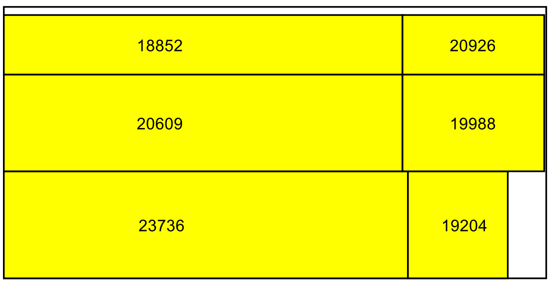
(j) B3 第 191 张板材(5-0218S)利用率为 0.9626



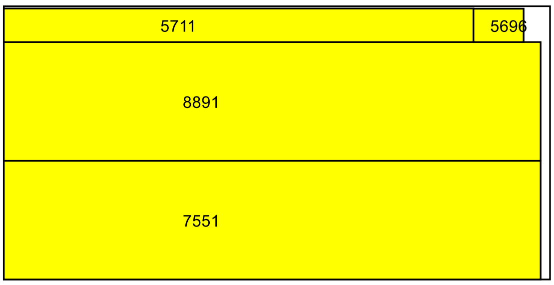
(k) B3 第 1745 张板材(GGXK-0418S)利用率为 0.9625



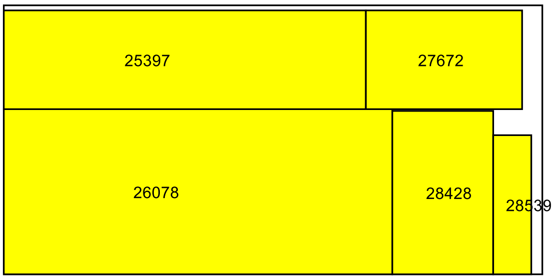
(l) B3 第 1373 张板材(JH-0215S)利用率为 0.9312



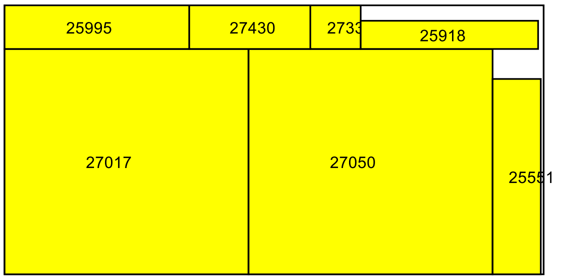
(m)B4 第 1584 张板材(NBSY)利用率为 0.9400



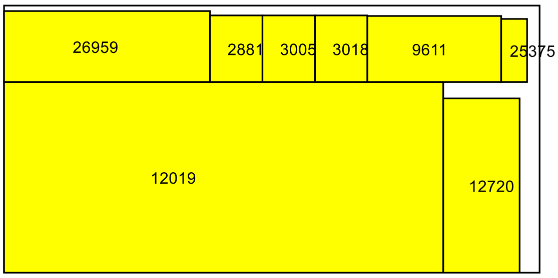
(n) B4 第 1110 张板材(JH-02185)利用率为 0.9708



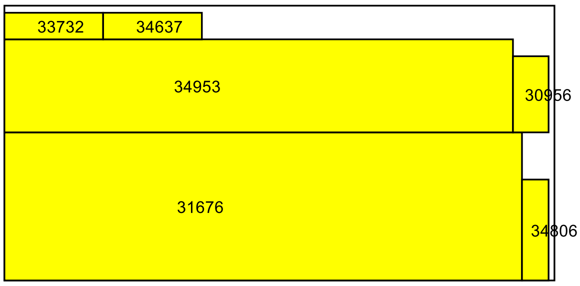
(o) B4 第 652 张板材(NHLP-0218S)利用率为 0.9467



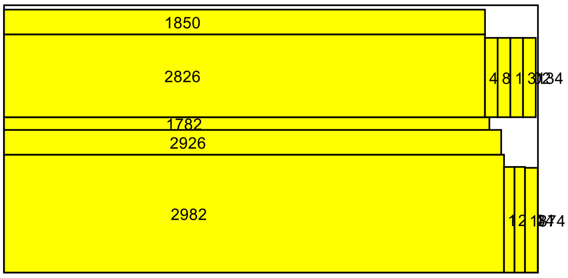
(p) B4 第 309 张板材(GDMY-0215S)利用率为 0.9650



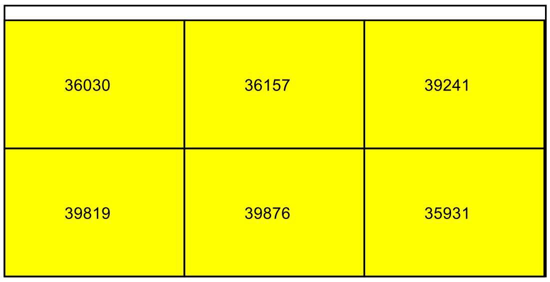
(q) B5 第 2006 张板材(HYBW-0218S)利用率为 0.9271



(r) B5 第 1630 张板材(CGS-0218S)利用率为 0.8906



(s) B5 第 1012 张板材(HT9-0215S)利用率为 0.9580



(t) B5 第 506 张板材(JH-02185)利用率为 0.9432

图 17 数据集 B 中随机抽取的排版方式

五、论文的改进与评价

本文主要采用了整数线性规划模型分别对方形件的下料问题和订单组批问题进行优化，针对优化的算法过程中对比了经典的二维三阶段排样算法，同时参考了【文献 3】中的整数线性规划模型，并对其中存在产品无法旋转问题进行改进，考虑了产品可以旋转问题。因此该模型的复杂度达到了 $7n^2+n$ 阶，即 $O(n^2)$复杂度，算法执行所需时间较长。故此，在程序执行的环节，我们可以考虑采用效率更高效的算法进行优化，这将会节省大量的时间和精力。

参考文献

[1] Silva E, Alvelos F, De Carvalho J M V. An integer programming model for two-and three-stage two-dimensional cutting stock problems[J]. European Journal of Operational Research, 205(3): 699-708, 2010.

[2] Cui Y, Huang B. Reducing the number of cuts in generating three-staged cutting patterns[J]. European journal of operational research, 218(2): 358-365, 2012.

[3] Puchinger J, Raidl G R. Models and algorithms for three-stage two-dimensional bin packing[J]. European Journal of Operational Research, 183(3): 1304-1327, 2007.

[4] 刘倩. “一刀切” 约束下的矩形件优化排样算法比较与整合研究[D]. 河北工业大学, 2012.

附录

%%子问题 1 中参考文献[3]中产品旋转与为旋转对比示意图

```
%% 测试
clear
clc
% H = [640 1800 500 700];
% W = [1220 520 700 1300];
B = [300 420 420 500 500 1180];
A = [2440 1740 700 800 400 500];
r = [1 1 1 1 1 1 1];
[~,Z] = aa(A,B);
[~,Z,r] = bb(A,B);
%% 绘图
nig = length(B);
IO1 = cell(nig,9);
Id_SIN = unique(Z(:,1));
for LON = Id_SIN(:)'
    figure, patch([0 1220 1220 0 0],[0 0 2440 2440 0],'white')
    hold on, axis('equal','off')
    X0 = 0;
    Y0 = 0;
    Z_Sin = Z(Z(:,1) == LON,:);
    id_tiao = unique(Z_Sin(:,2));
    for k = id_tiao(:)'
        a0 = 0;
        Z_tiao = Z_Sin(Z_Sin(:,2) == k,:);
        id_zhan = unique(Z_tiao(:,3));
        for jaa = id_zhan(:)'
            Z_zhan = Z_tiao(Z_tiao(:,3) == jaa,:);
            Cc = rand(1,3);
            for m = 1:size(Z_zhan,1)
                id_chanpin = Z_zhan(m,4);
                r_i = r(id_chanpin);
                b_i = B(id_chanpin);
                a_i = A(id_chanpin);
                A_i = a_i*r_i+b_i*(1-r_i);
                B_i = b_i*r_i+a_i*(1-r_i);
                IO1{id_chanpin,2} = find(Id_SIN == LON);
                IO1{id_chanpin,4} = X0;
                IO1{id_chanpin,5} = Y0;
                IO1{id_chanpin,6} = B_i;
                IO1{id_chanpin,7} = A_i;
                IO1{id_chanpin,8} = X0+B_i;
```

```
                IO1{id_chanpin,9} = Y0+A_i;
                X = [X0 X0+B_i X0+B_i X0 X0];
                Y = [Y0 Y0 Y0+A_i Y0+A_i Y0];
                patch(X,Y,Cc,'FaceAlpha',0.5)
                Y0 = Y0+A_i;
            end
            X0 = max(cell2mat(IO1(Z_zhan(:,4),8)));
            Y0 = min(cell2mat(IO1(Z_zhan(:,4),5)));
            z0 = max(cell2mat(IO1(Z_zhan(:,4),9)));
            a0 = max([z0 a0]);
        end
        X0 = 0;
        Y0 = a0;
    end
end
%%函数
function [num_tt,R,A,B] = aa(A,B)

%%
A = A(:)';
B = B(:)';
na = length(A);
Is = eye(na);
a_max = 2440;
b_max = 1220;
idx_1a = 0*na^2+1:1*na^2;
idx_2b    = 1*na^2+1:2*na^2;
idx_3c = 2*na^2+1:3*na^2;
na_vars        = 3*na^2;

%%
ff = zeros(1,na_vars);
for l1 = 1:na
    ff_la = kron(Is(l1,:),Is(l1,:));
    ff(idx_3c) = ff(idx_3c)+ff_la;
end

%%
Hs_naeq = cell(1,0); Hs_eq = cell(1,0);
Ws_nAeq = cell(1,0); Ws_eq = cell(1,0);

%%
Aa_eq = zeros(na,na_vars);
for ia1 = 1:na
```

```
    cs_i = [ones(1,ia1) zeros(1,na-ia1)];
    ap_i = kron(Is(ia1,:),cs_i);
    Aa_eq(ia1,idx_1a) = ap_i;
end
bk_eq = ones(na,1);
Hs_eq{length(Hs_eq)+1} = Aa_eq;
Ws_eq{length(Ws_eq)+1} = bk_eq;

%%
Aa_eq = zeros(na,na_vars);
for jaa = 1:na
    ap_jaa_1 = -kron(Is(jaa,:),Is(jaa,:));
    ap_jaa_2 = kron(Is(jaa,:),[ones(1,jaa) zeros(1,na-jaa)]);
    Aa_eq(jaa,[idx_1a idx_2b]) = [ap_jaa_1 ap_jaa_2];
end
bk_eq = zeros(na,1);
Hs_eq{length(Hs_eq)+1} = Aa_eq;
Ws_eq{length(Ws_eq)+1} = bk_eq;

%%
i1dx = [idx_1a idx_2b];
A_naeq = zeros(na-1,na_vars);
for jaa = 1:na-1
    ap_jaa_1 = kron([zeros(1,jaa-1) A(jaa:na)],Is(jaa,:));
    ap_jaa_2 = -kron(Is(jaa,:),[A(1:jaa) zeros(1,na-jaa)]);
    A_naeq(jaa,i1dx) = [ap_jaa_1 ap_jaa_2];
end
W_naeq = zeros(na-1,1);
Hs_naeq{length(Hs_naeq)+1} = A_naeq;
Ws_nAeq{length(Ws_nAeq)+1} = W_naeq;

%%
A_naeq = zeros(na-1,na_vars);
for kbb = 1:na-1
    aa_k_1 = kron([zeros(1,kbb-1) B(kbb:na)],Is(kbb,:));
    aa_k_2 = b_max*kron(Is(kbb,:),Is(kbb,:));
    A_naeq(kbb,idx_2b) = aa_k_1-aa_k_2;
end
W_naeq = zeros(na-1,1);
Hs_naeq{length(Hs_naeq)+1} = A_naeq;
Ws_nAeq{length(Ws_nAeq)+1} = W_naeq;

%%
Aa_eq = zeros(na,na_vars);
```

```
for kbb = 1:na
	aa_k_1 = -kron(Is(kbb,:),Is(kbb,:));
	aa_k_2 = kron(Is(kbb,:),[ones(1,kbb) zeros(1,na-kbb)]);
	Aa_eq(kbb,[idx_2b idx_3c]) = [aa_k_1 aa_k_2];
end
bk_eq = zeros(na,1);
Hs_eq{length(Hs_eq)+1} = Aa_eq;
Ws_eq{length(Ws_eq)+1} = bk_eq;

%%
A_naeq = zeros(na-1,na_vars);
for l1 = 1:na-1
	aa_l_1 = kron([zeros(1,l1-1) A(l1:na)],Is(l1,:));
	aa_l_2 = a_max*kron(Is(l1,:),Is(l1,:));
	A_naeq(l1,idx_3c) = aa_l_1-aa_l_2;
end
W_naeq = zeros(na-1,1);
Hs_naeq{length(Hs_naeq)+1} = A_naeq;
Ws_nAeq{length(Ws_nAeq)+1} = W_naeq;

%%
A_naeq = cat(1,Hs_naeq{:}); Aa_eq = cat(1,Hs_eq{:});
W_naeq = cat(1,Ws_nAeq{:}); bk_eq = cat(1,Ws_eq{:});

%%
IO = zeros(1,na_vars);
FO = ones(1,na_vars);
QE = triu(ones(na));
FO(idx_2b) = QE(:)';
FO(idx_3c) = QE(:)';
for jaa = 1:na-1
	for ia1 = jaa+1:na
		if (B(ia1) ~= B(jaa)) || (A(ia1)+A(jaa) > a_max)
			QE(jaa,ia1) = 0;
		end
	end
end
FO(idx_1a) = QE(:)';

%%
i1dx = 1:na_vars;
optA1 = optimoptions('intlinprog','Display','final');
[ax1,num_tt] = intlinprog(ff,i1dx,A_naeq,W_naeq,Aa_eq,bk_eq,IO,FO,optA1);
```

```
%%
num_tt = round(num_tt);
la = reshape(ax1(idx_1a),na,na);
b2 = reshape(ax1(idx_2b),na,na);
c3 = reshape(ax1(idx_3c),na,na);
la = round(la*100)/100;
b2 = round(b2*100)/100;
c3 = round(c3*100)/100;

%%
na = length(B);
R = nan(na,4);
for ia1 = 1:na
        jaa = find(la(:,ia1));
        kbb = find(b2(:,jaa));
        l1 = find(c3(:,kbb));
        R(ia1,:) = [l1 kbb jaa ia1];
end

%%子问题 1 算法（主程序，相关函数在附件）
clc;clear;format compact
clear; clc; close all; tic;
  A=xlsread('Num1.xls');t0=size(A,1)
  meipaihaobancai=1:t0;      %没排好的板材数量
tc0=-1;
  JieGuo=[];;LiYong=[];
  JieGuo1=[];
  LiYong1=[]; suanfaxuanze=1
  TT=[];
  for m0=1:130
          if isempty(meipaihaobancai)
                    break
            end
          figure(m0);hold on;axis equal;axis off;
            chanpin0=1;      %第一个产品项
            width0=1220;length0=2440
            if suanfaxuanze==0
[A,meipaihaobancai,width0,length0,JieGuo,LiYong,m0,tc0,liyong]=function_plot(A,meipaihaobancai,width0,length0,JieGuo,LiYong,m0,tc0)
            end
            if suanfaxuanze==1
                    width0=1220;length0=2440
```

```
[A1,meipaihaobancai1,width0,length0,JieGuo1,LiYong1,m01,tc01,liyomg1]=function_00(A,meipaihaobancai,width0,length0,JieGuo,LiYong,m0,tc0)
                    width0=2440;length0=1220

[A2,meipaihaobancai2,width0,length0,ST2,Useb2,m02,tc02,liyomg2]=function_00(A,meipaihaobancai,width0,length0,JieGuo,LiYong,m0,tc0)
                    if liyomg1>=liyomg2
                           width0=1220;length0=2440

[A,meipaihaobancai,width0,length0,JieGuo,LiYong,m0,tc0,liyong]=function_plot(A,meipaihaobancai,width0,length0,JieGuo,LiYong,m0,tc0)
                    else
                             TT=[TT m0];width0=1220;length0=2440;

[A,meipaihaobancai,width0,length0,JieGuo,LiYong,m0,tc0,liyong]=function_plotww(A,meipaihaobancai,width0,length0,JieGuo,LiYong,m0,tc0)

                    end
           end
           pause(eps)
           close all
  end
  clc
  Zongtiliyong=sum(A(:,4).*A(:,5))/((m0-1)*1220*2440)
  fprintf(' 使用板材的数量为：%d   张，板材的总体利用率:%f\n ',tc0+1,Zongtiliyong)
  disp('竖排 的编号为：')
  TT
    disp('每张板材的利用效率分别为：')
    LiYong
      xlswrite('LiYong.xls',LiYong)
xlswrite('JieGuo.xls',JieGuo)
    save mydataA01
%%子问题 2 算法（主程序，相关函数在附件）
clear, clc, close all
[Data,Txt] = xlsread('dataB1.csv');

Len = Data(:,4);
Wid = Data(:,5);

caizhi = Txt(2:end,2);
caizhi_uniq = unique(caizhi);
h = nan(length(caizhi),1);
for i = 1:length(caizhi_uniq)
       h(contains(caizhi,caizhi_uniq{i})) = i;
```

```
end

OO = Txt(2:end,6);
OO_uniq = unique(OO);
oo = nan(length(OO),1);
for i = 1:length(OO_uniq)
	oo(contains(OO,OO_uniq{i})) = i;
end

suijishu = zeros(length(OO_uniq),length(caizhi_uniq),2);
for i = 1:length(OO_uniq)
	for j = 1:length(caizhi_uniq)
		idx = (oo == i & h == j);
		suijishu(i,j,1) = sum(idx);
		suijishu(i,j,2) = sum(Len(idx).*Wid(idx));
	end
end

Data(:,2)=h;
Data(:,6)=oo;



save dataB1 Data caizhi h OO oo suijishu

suijishu_r1=suijishu(:,:,1);suijishu_s1=suijishu(:,:,2);
suijishu_r1=suijishu(:,:,1);suijishu_s1=suijishu(:,:,2);
disp('每个材质总 de 数量与面积')
S1=[1:size(suijishu_r1,2);sum(suijishu_r1);sum(suijishu_s1)]

disp('每个订单总 de 数量与面积')
S2=[1:size(suijishu_r1,1);sum(suijishu_r1');sum(suijishu_s1'/10^6)]
S3=corrcoef(suijishu_r1');
for m=1:size(S3,1)
	Cork=S3(:,m);
	[B mc]= sort(Cork,'descend');
	m2=find(B<=0.6);
	FL{m}=mc(1:m2)';
end
celldisp(FL)

for m1=1:size(S3,1)
	Fkc=FL{m1};DDta=Fkc
	for m2=2:length(Fkc)
```

```
            DDta=[DDta FL{Fkc(m2)}];
        end
        FC{m1}=unique(DDta,'stable');
end
meipaihaobancai=1:size(S3,1);
t0=1;
[mc mc]= sort(S2(3,:),'descend');
S4=S2(:,mc);

n00=0;
for m=1:size(S3,1)
        mc=S4(1,m); Ck=FC{mc};
        if ismember(mc,meipaihaobancai)
              suanfaxuanze=ismember(Ck,meipaihaobancai);m=find(suanfaxuanze==1);
              zzbp=Ck(m); AX=S2([2 3],zzbp);
              if size(AX,2)>1
                    DDta=sum(AX');
              elseif size(AX,2)==1
                      DDta=AX';
              else
                      DDta=[0 0];
              end


            if DDta(1)>=100 | DDta(2)>=250
                    zb=1;t0=length(zzbp);
                    while DDta(1)>=1000 | DDta(2)>=250
                            t0=t0-1;
                            DDta=sum(S2([2 3],zzbp(1:t0))')
                    end
                      n00=n00+1;kcc=zzbp(1:t0);
                      meipaihaobancai=setdiff(meipaihaobancai,kcc);
                      FZ{n00}=kcc;
                      SJ(:,n00)=[n00 length(kcc) DDta];
            end
        end
end
celldisp(FZ)
SJ
%排序
      [mc mc]= sort(SJ(4,:),'descend');
      SJD=SJ(:,mc)
%
```

```
    SJJD=SJD;

  t0=size(SJ,2);
  DDta=[];FZC=FZ;
    for m1=t0:-1:1
        As1=SJJD(:,m1) ;
        for m2=1:m1-1
            As2=SJJD(:,m2);
            As12=As1+As2;
            if As12(3)<=1000 & As12(4)<=250
                SJJD([2:4],m2)=As12(2:4);
                DDta=[DDta;m1 m2];
                FZC{As2(1)}=[FZ{As2(1)}   FZ{As1(1)}];
                break
            end
        end
    end
    DDta
    t0=DDta(end,1)-1;
     for m=1:t0
       NewFangFa{m}=FZC{SJJD(1,m)};
    end
   celldisp(NewFangFa)

    fprintf('一共分成%d 组\n',t0)
    NewFangZhi=[1:t0 ;SJJD(:,1:t0)]
%
  for m1=1:t0
      SetS=NewFangFa{m1}
      Setm=[];
      for m2=1:length(SetS)
          m=find(Data(:,6)==SetS(m2));
          Setm=[Setm;Data(m,:)];
      end
      SetPX{m1}=Setm;
      if m1<=9
         string=['Nb0' num2str(m1) '.xls'];
      else
         string=['Nb' num2str(m1) '.xls'];
      end
      xlswrite(string,Setm)
  end
  save mydate01
```