**五一数学建模竞赛**

**承 诺 书**

我们仔细阅读了五一数学建模竞赛的竞赛规则。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与本队以外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的, 如果引用别人的成果或其它公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们愿意承担由此引起的一切后果。

我们授权五一数学建模竞赛组委会，可将我们的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等）。

参赛题号（从A/B/C中选择一项填写）： B

参赛队号： B19124367

参赛组别（研究生、本科、专科、高中）： 本科

所属学校（学校全称） 南京理工大学

参赛队员（打印并签名）： 1. 许晓明

2. 陈梦佳

3. 陈靓颖

日期： 2019 年 5 月 4 日

获奖证书邮寄地址： 江苏省南京市玄武区孝陵卫街200号南京理工大学 邮政编码： 210094

收件人姓名： 许晓明 联系电话： 18851196872

**五一数学建模竞赛**

****

**题 目**

**关键词：动态规划模型**

**摘 要：**

一、问题重述

1.1问题背景

随着现代机械加工工业的发展，无论是在大中型制造企业还是传统加工企业，对玻璃、金属、木板、塑料、面料等原材料的切割加工问题涉及到工艺、家具、电子、装饰等众多领域。为了减少生产成本，增加企业利润，就需要通过更合理的切割方式来提高原料的利用率，尽可能的减少下料。如若仅凭借个人经验排料则会导致原材料利用率低、工作效率低下、成本增加。因此我们需要通过已知的板材数据和企业生产计划来制定合理高效的下料切割计划、进行板材定额及生产成本计算，为材料库准备材料提供计划，为企业经管部门提供决策依据和材料采购预算。

1.2需要解决的问题

徐州一家具厂新进了一批木板原材料，已知板材原料S1及各待加工产品的信息，忽略木板厚度和切割缝宽，编制出与生产任务和利润相匹配的板材切割列表。

（1）在一块S1上只切割P1，求木板利用率最大时切割的P1数量及该最大利用率值。

（2）在一块S1上同时切割P1、P3，求木板利用率按从高到低顺序排名前三的切割方案及利用率值。

（3）在完成表二中P1、P3生产任务的基础上，求出木板总利用率最高时的切割方案及利用率值。

（4）在完成表二中P1、P2、P3、P4生产任务的基础上，求出木板总利用率最高时的切割方案及最大总利用率值。

（5）在100张S1上切割P1、P2、P3、P4，求所获利润最大时的切割方案及总利用率值。

1. 模型假设
2. 假设
3. 符号说明

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

四、问题分析与求解

4.1第一、第二问的问题分析与求解

4.1.1问题分析

设矩形木板S1长度为L，宽度为W，根据题目所给条件已知L=3000mm，W=1500mm。以S1为原材料，问题一在一块矩形木板S1上切割面积更小的矩形产品{P1},问题二在一块矩形木板S1上切割面积更小的矩形产品{P1，P3}。设该矩形产品的长度为，宽度为，切割n个小矩形产品，则木板利用率。问题一只在木板S1上切割P1产品，问题二在木板S1上同时切割P1、P3两种产品，由于二者都只是在一块木板上进行切割，都可简化为在二维平面上的矩形数量分配问题，也都是为了求得木板利用率最高时产品的数量，因此我们对一、二问用同一模型、算法进行分析求解。

由于假设忽略切割木材过程中的损耗，可将该切割问题转化成小矩形产品在大矩形原材料上的排列问题，通过二维坐标的建立，我们建立一个动态规划模型，又将其转化为在一定约束条件下求最优解问题。继而通过能得到局部最优解的剩余矩形法与能得到全局最优解的模拟退火算法相结合，计算得到最大木板利用率时的板材切割数量。

4.1.2 动态规划模型的建立

由于忽略切割过程的木材损耗，我们能将一块木板原材料的切割过程转换成小木板在木板原材料上的排样过程，这是一个复杂的组合优化问题。我们引入动态规划模型，每一个小矩形产品在板材上排列后又会产生剩余的未排列矩形区域，设每一个即将被排列的矩形产品在木板原材料上排列时左下角的横纵坐标分别为，则可在矩形板材的每一次余料上建立一个不断被刷新的动态二维坐标。

对于一种矩形产品在木板原材料上的排列有两种方式：一是产品的长与原材料的长平行，产品宽与原材料的宽平行；二是产品的长与原材料的宽平行，产品宽与原材料的长平行。

以方式一排列：

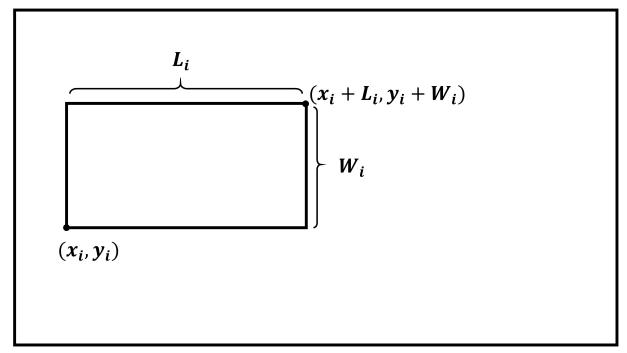


图1：长平行长排列的坐标图



以方式二排列：

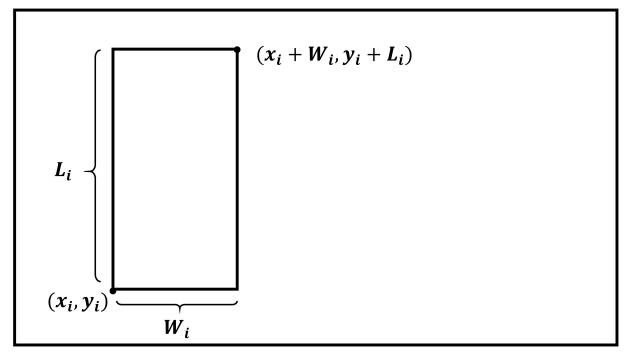


图2：长平行宽排列的坐标图



因此，我们将矩形产品在原材料上的排列问题转化为约束条件下求的最优解问题。

4.1.3 算法的介绍

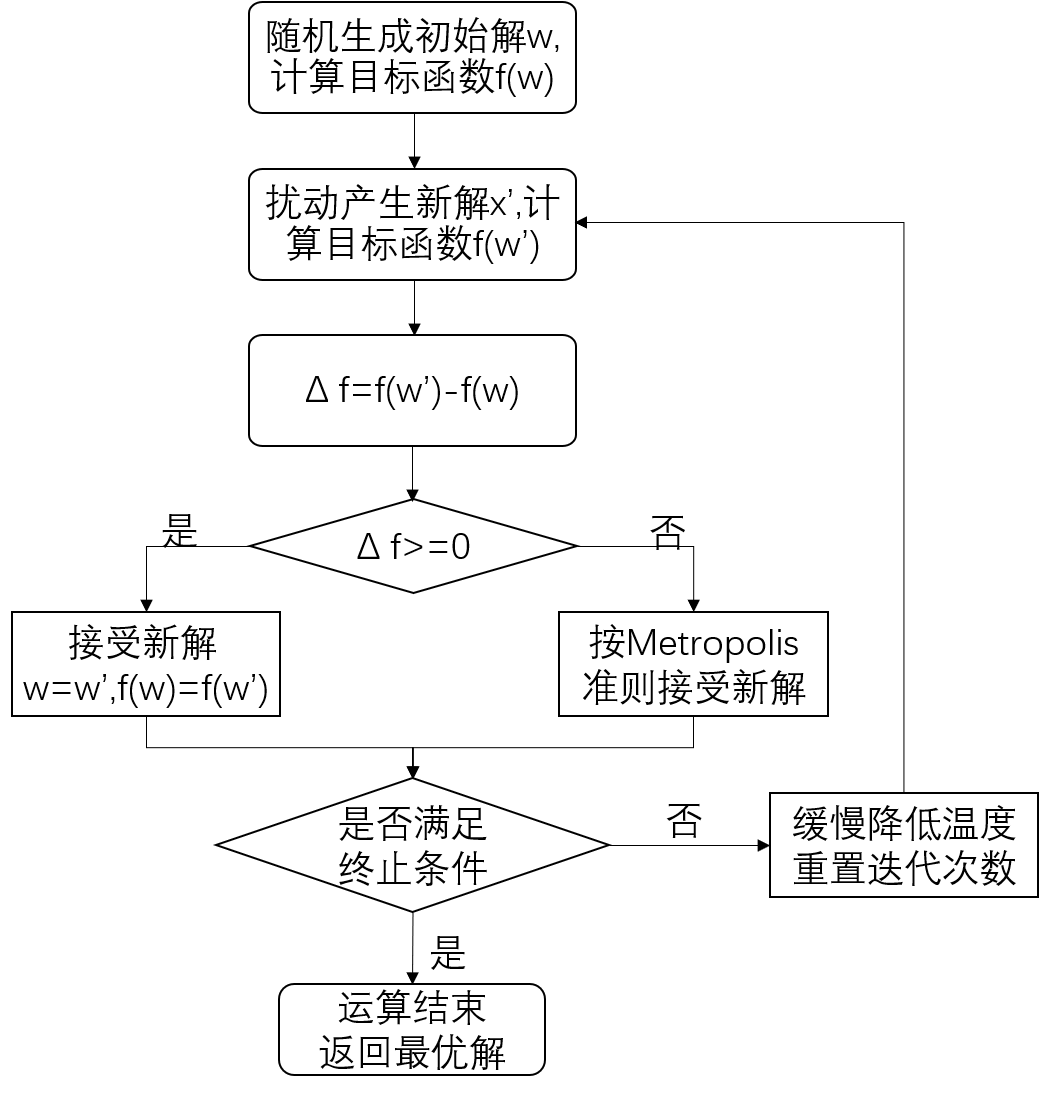
首先采取“剩余矩形匹配算法”来计算该最优解。假设有n个矩形产品，不同种类的产品在S1上排列时又在不断地产生新的小矩形，计算产生的剩余矩形数据，判断图形关系，进行新的待排矩形产品与剩余矩形板材的匹配运算。于是不断地往新产生的小矩形中继续填充矩形产品，直到剩余矩形无法继续排列矩形产品为止。[1]剩余矩形匹配算法是一种在局部寻求最优解的策略，它的条件为:

（1）排列时任意两个矩形之间不能有重叠；

（2）矩形产品不能排列到板材原材料之外；

（3）保证矩形产品在板材上已不能再排列下去；

而模拟退火算法是一种在全局寻求最优解的策略，结合剩余矩形匹配算法的局部寻优特点，我们对剩余矩形匹配算法和模拟退火算法融合进行优化后的算法框图如下：



图三：模拟退火剩余矩形算法流程图

将函数作为目标函数，则目标函数值越大，木板利用率越高。先初始化，给定初始温度随机生成初始解，计算目标函数值。随后对当前解随机扰动产生新解，计算新解对应的函数值。若新解对应的函数值大于旧解，则接受新解为当前解。不断重复上述步骤，迭代N次后若算法满足终止条件，则输出当前解为最优解；若不满足终止条件则缓慢降低当前温度再重复上述步骤。

降温方式对模拟退火算法优化结果有很大影响。如果温度下降过快，可能会丢失极值点；反之，则大大降低算法的收敛速度。本文定义的降温公式为[3]



其中 ＞0为初始温度，T（k）为当第k次迭代的温度，m为大于等于1的常数。当满足下列条件时算法终止:

1. 温度达到终止温度时；
2. 当满足约束条件时。

4.1.4答案求解

问题一的结果：

|  |  |
| --- | --- |
| P1的数量 | 木板利用率 |
| 59个 | 98.298% |

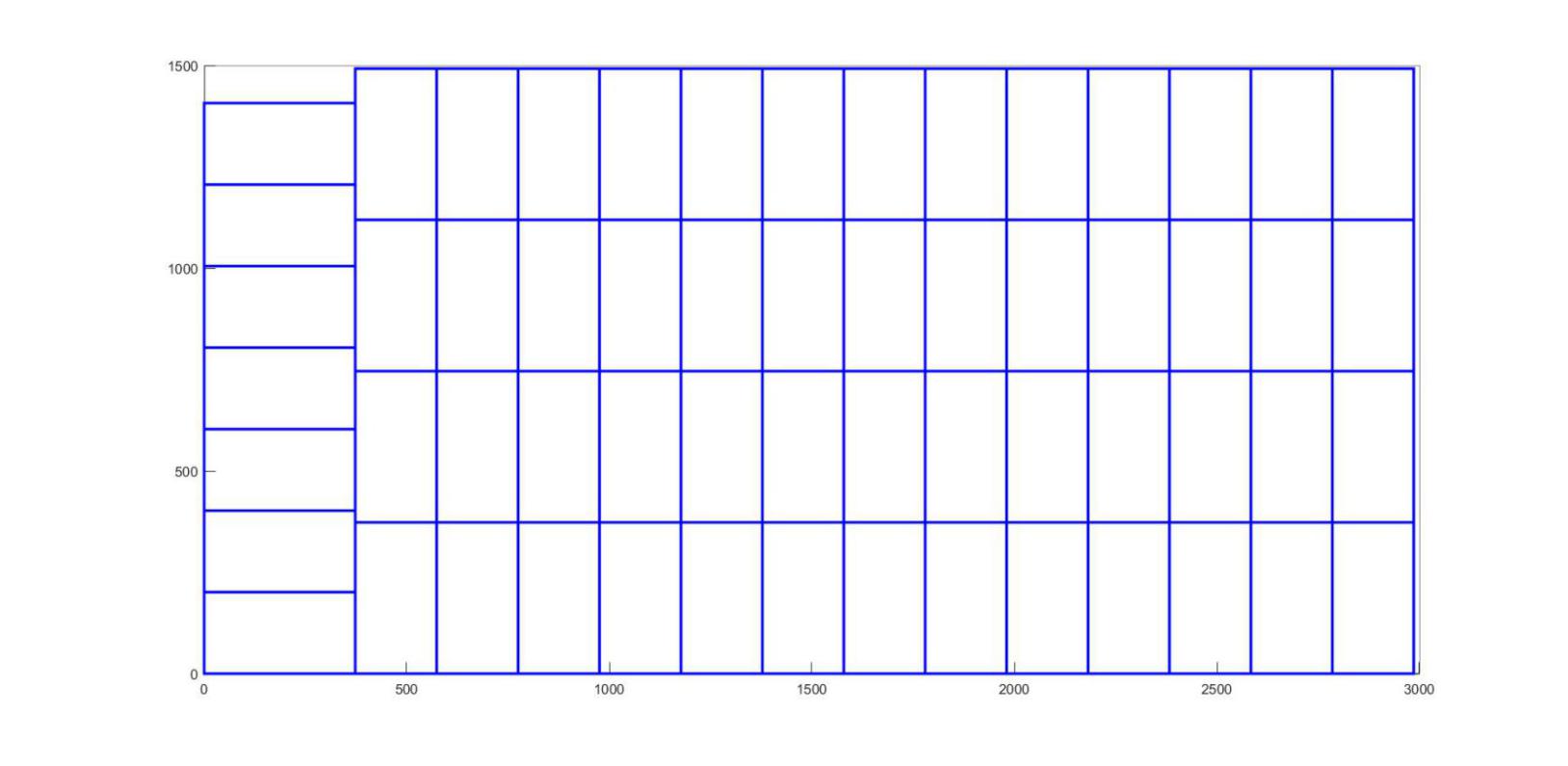


图4：问题一的切割方案图

问题二的结果：

|  |  |  |  |
| --- | --- | --- | --- |
| 方案编号 | P1的数量 | P3的数量 | 木板利用率 |
| 1 | 44 | 12 | 98.100% |
| 2 | 45 | 11 | 97.700% |
| 3 | 35 | 19 | 97.568% |

4.2第三、第四问的问题分析与求解

4.2.1问题分析

4.3

4.4

4.5

参考文献

[1]

附录

1. 问题一、二代码如下

clear

board\_size = [3000, 1500];

%Primitive arrangement of product dimensions

S0=zeros(60+48,2);

for i=1:60

S0(i,1)=373;

S0(i,2)=201;

end

% for j=48:48+60

% S0(i,1)=406;

% S0(i,2)=229;

% end

% constants

% Temperature drop using Lundy and Mess formula

d\_0 = 1; %According to the definition of f

K = 10;

t0 = K \* d\_0; %initial temperature

tf = 10;

M = 50000; %Upper limit of total iteration times

beta = (t0-tf) / (M\*t0\*tf);

%

L = 200; %Lower bound of iteration times at the same temperature

U = 500; %The Upper Bound of iteration times at the Same Temperature

Accept\_Ratio = 0.5;

%Ratio of the number of iterations accepted to the number of iterations at the same temperature

S = S0; % Initial solution

i\_Iter = 2; % times of iterations at the same temperature

nTotal\_Iter = 0; % Total times of iterations

k = 1; % times of iterations at different temperatures

ratio(k) = 1; % Cost function: ratio of surplus to total consumption

tk(1) = t0; % Temperature

fmin = 1; % Minimum Cost Function

solution = S; % optimum solution

while( nTotal\_Iter<M && fmin > 0.02 )

i\_Iter = 0;

iAccept\_Iter = 0;

fS = f( S, board\_size, false );

while ( i\_Iter<U )

Sstar = N( S );

df = 0;

fStar = f( Sstar, board\_size, false);

df = fStar - fS;

if ( df <=0 || exp( -df / tk(k) ) > rand )

S = Sstar;

fS = fStar;

iAccept\_Iter = iAccept\_Iter + 1;

end

% When the temperature drops to the limit

if fS < fmin

fmin = fS;

solution = S;

end

i\_Iter = i\_Iter + 1;

nTotal\_Iter = nTotal\_Iter + 1;

if ( i\_Iter > L && iAccept\_Iter/i\_Iter > Accept\_Ratio )

fprintf('iAcceptIter/iIter = %f\n', iAccept\_Iter/i\_Iter);

break; %Temperature drop

end

end

ratio(k) = f(S, board\_size, false);

fprintf('iOuterLoop = %d, iIter = %d, tk = %f, ratio = %f, fbestratio=%f\n', k, i\_Iter, tk(k), ratio(k), fmin);

%tk(k+1) = tk(k) / ( 1+beta\*tk(k) );

%Temperature drop

tk(k+1) = 0.9 \* tk(k);

k = k + 1;

end

fmin = f(solution, board\_size, true);

fprintf('best ratio = %f, nTotalIter = %d\n', fmin, nTotal\_Iter);