

摘要：本文讨论了有交货时间限制的大规模单一原材料下料问题。对于一维下料问题，本文提出一种新的算法：DP 贪婪算法。在一维的基础上建立了二维的求解模型，运用降维思想结合一维的 DP 贪婪算法，给出解决该模型的算法。数值计算结果表明该算法对大规模下料问题是有效的。

关键词：下料问题，DP，贪婪算法

1、问题描述

单一原材料下料问题。设这种原材料呈长方形，长度为 L ，宽度为 W ，现在需要将一批这种长方形原料分割成 m 种规格的零件，所有零件的厚度均与原材料一致，但长度和宽度分别为 $(l_1, w_1), \dots, (l_m, w_m)$ ，其中 $w_i < l_i < L, w_i < W, i=1, \dots, m$ 。 m 种零件的需求量分别为 n_1, \dots, n_m 。下料时，零件的边必须分别和原材料的边平行。这类问题在工程上通常简称为二维下料问题。特别当所有零件的宽度均与原材料相等，即 $w_i = W, i=1, \dots, m$ ，则问题称为一维下料问题。

一个好的下料方案是在生产能力容许的条件下，以最少数量的原材料，尽可能按时完成需求任务，同时下料方式数也尽量地小。

2、一维下料问题

2.1 模型假设

在充分了解并分析了实际情况后，我们对一维下料问题提出如下假设：

- (1) 每天下料的数量受到企业生产能力的限制，在未完成需求任务前，每天下料的数量等于最大下料能力。
- (2) 每个切割点处由于锯缝所产生的损耗不可忽略。
- (3) 增加一种下料方式大致相当于使原材料总损耗增加 0.08%。
- (4) 每种零件有各自的交货时间，若某零件无交货时间，则记该零件交货时间为无穷大。

2.2 一维单一原材料实用下料问题的模型

根据公司要求，目标是既要所用材料最少，也要下料方式少。记 m ：零件种类总数， x_i ：第 i 种下料方式下料的根数， k ：下料方式的种类数， δ_i ：第 i 种下料方式的余料。借助函数

$signal(x_i) = \begin{cases} 1 & x_i > 0 \\ 0 & x_i = 0 \end{cases}$ ，可得所用材料 $f_1(x)$ 和采用的下料方式 $f_2(x)$ 分别为：

$$f_1(x) = \sum_{i=1}^k \delta_i x_i, \quad f_2(x) = \sum_{i=1}^k signal(x_i)$$

借助模型假设中假设 (3)：增加一种下料方式大致相当于使原材料总损耗增加 0.08%。故可将双目标转化为单目标：

$$\min f(x) = \sum_{i=1}^k \delta_i x_i \times \left(1 + 0.08\% \times \left(\sum_{i=1}^k \text{signal}(x_i) - 1 \right) \right)$$

由于每天下料的数量受到企业生产能力 \max 的限制, 假设在 d 天内各种下料方式的下料总根数分别为 y_1, y_2, \dots, y_k , 零件 j 的需求数量为 n_j , 第 i 种下料方式下料一次产生的零件 j 的个数为 a_{ij} 。设 F 是要求在 d 天内完成的零件集合, 则必须满足:

$$\begin{cases} \sum_{i=1}^k a_{ij} y_i = n_j & (j \in F) \\ \sum_{i=1}^k y_i \leq d \cdot \max \end{cases}$$

即 d 天内需要完成的零件必须在前 $d \cdot \max$ 根原材料的切割中得到。

根据上述分析, 得到有时间限制的一维单一下料问题模型:

$$\begin{aligned} \min f(x) &= \sum_{i=1}^k \delta_i x_i \times \left(1 + 0.08\% \times \left(\sum_{i=1}^k \text{signal}(x_i) - 1 \right) \right) \\ \text{s. t. } &\begin{cases} \sum_{i=1}^k a_{ij} x_i = n_j & (j = 1, \dots, m) \\ \sum_{i=1}^k y_{i,d_j} \leq d_j \cdot \max & (j = 1, \dots, m) \\ \sum_{i=1}^k a_{ij} y_{i,d_j} = n_j & (j \in S_{d_j}) \\ y_{i,d_l} \leq y_{i,d_j} \leq x_i & (i = 1, \dots, k; d_l < d_j) \\ x_i, y_{i,d_j} \geq 0 \text{ 且为整数} & (i = 1, \dots, k; j = 1, \dots, m) \end{cases} \end{aligned}$$

其中 y_{i,d_j} : 第 i 种下料方式前 d_j 天内下料总根数, S_{d_j} : 交货时间均等于 d_j 的零件集合

2.3 模型求解

对于该问题, 因为可能的下料方式将随需要的零件种类数量成指数级增长, 所以它是一个 NP-Hard 问题。这样对于大多数问题, 一般方法无法得到最优结果或无法及时得到最优结果。因此对于大规模的一维下料问题, 我们给出了结合动态规划和贪婪算法的新算法, 称之为 DP 贪婪算法。

基本思想是: 对模型计算时, 不用先得到一定数量的下料方案, 而是在选取下料方案时就以数学模型中的目标和约束条件为基础来进行寻找。

为了保证尽量节省材料, 应该尽量将比较大的零件先进行处理, 并同时辅以长度小的零件, 以保证单个原料的利用率尽量大。因此对每一个零件按照其长度大小依次给定处理顺序的权值。为了保证时间的要求, 有要求的零件应该尽量优先处理, 对每一个零件按时间紧迫度 t 依次给定一个处理顺序的权值。两者的结合将作为每一个零件动态规划初始权值。在决定了处理顺序后, 首先利用贪婪思想, 选取当前尚没有得到的零件集合中权值最大的一个进行处理。调用动态规划方法, 得到一种下料方式, 此方法里含有当前的零件, 在得到此下料方式后, 先尽可能按照此方式进行处理, 以尽量减少下料方式数, 然后再应用贪婪思想。依次类推, 直到得到所有的零件。这样我们将得到一种下料方案。如果此方案满足约束要求则停止处理, 否则对权值进行调整, 如

果结果不能满足时间紧迫度的限制，则将优先权值步长直接调节到理论上限，随后通过二分查找的方法进行选择，如果材料利用率过低，则参照以上方法进行调节。而后重复上述过程，直到得到合理结果。

算法描述：

1. 局部最优

//计算当前单根利用率最大值，并得到一组可行下料方案

FOR I = 1 TO 工件总种类数

FOR J = 原材料总长度 DOWNT0 0

IF 在 J 的位置已经有解

FOR K = 第 I 件工件中未切割的数量 DOWNT0 1

当前长度 = J + 第 I 件工件的长度*K

IF 当前长度位置尚未得到解 THEN

保存当前解

ELSE 对两个解进行比较选取较优解

FOR I = 材料长度 DOWNT0 1

IF 当前长度有解存在 THEN

返回解

2. 全局贪婪

对所有需要的零件进行处理

FOR I = 1 TO 工件种类总数

WHILE 如果当前种类还有剩余(按照权值大小依次处理) DO

利用上述局部最优处理选取一种至少含有当前种类一根的最优解

累加计算结果

更新数据表格

3. 反复调整

调整权值

IF 得到全局的解法不合理

IF 不能按时完成零件 按规则加大优先权值

ELSE 浪费过于大 按规则加大长度权值

调用上述全局贪婪

3、二维下料问题

二维情况下，假设在矩形原料切割时采用正交切割，切割时的锯缝可以是直的也可以是弯的；不允许零件旋转；而且切割所引起的锯缝损耗忽略不计。

3.1 模型建立

假设共有 k 种不同下料方式，第 i 种下料方式下料的总块数记为 x_i ($i=1, \dots, k$)，并记第 i 种

下料方式产生零件 j 的个数记为 a_{ij} (如果某零件 j ($1 \leq j \leq M$) 满足 $l_j > L$ 或 $w_j > W$ ，则

$a_{ij} \equiv 0$ ($i=1, \dots, k$)，记第 i 种下料方式下料一次产生的余料（即料头）为 δ_i ($i=1, \dots, k$)。

二维单一原材料下料模型的建模思想与一维单一原材料下料模型类似。得到如下有交货时间限制的二维下料问题的数学模型：

$$\min f(x) = \sum_{i=1}^k \delta_i x_i \times \left(1 + 0.08\% \times \left(\sum_{i=1}^k \text{signal}(x_i) - 1 \right) \right)$$

$$\text{s. t.} \begin{cases} \sum_{i=1}^k (a_{i,j+m} + a_{ij}) x_i = n_j & (j=1, \dots, m) \\ \sum_{i=1}^k y_{i,d_j} \leq d_j \cdot \max & (j=1, \dots, m) \\ \sum_{i=1}^k a_{ij} y_{i,d_j} = n_j & (j=1, \dots, m) \\ y_{i,d_l} \leq y_{i,d_j} \leq x_i & (i=1, \dots, k; d_l < d_j) \\ x_i, y_{i,d_j} \geq 0 \text{ 且为整数} & (i=1, \dots, k; j=1, \dots, m) \end{cases}$$

3.3 模型求解

目前，解有交货时间限制的二维下料问题的常用方法是启发式算法，但是这种方法在大规模的下料问题中并不能将问题的规模降到一个合理的范围。

对于大规模的二维下料问题，本文给出新的求解方法。先利用降维思想将二维下料问题化为两个一维下料问题，对每一个一维下料问题，再使用本文一维下料问题的 DP 贪婪算法进行计算，再将两者的结果结合起来，得到最终的结果。

本文采用的降维思想为：第一步，先考虑长度（或宽度）这一维（以下采用先考虑宽度为例进行说明），将宽度相同的零件归为一类，对每一类，假设各自存在与该类等宽与原母板等长的母板。这样，每一类零件宽度与各自的母板宽度相等，这就转化为一维下料问题。故可借助一维下料模型的算法解出原母板在长度维上的切割方式。这种方式找到的是长度维上的局部近似最优。第二步，考虑宽度（或长度）这一维。由上一步，我们可以得到每一宽度各自所需的母板根数，可将每一类宽度视为一维切割中一个零件的长度，将每一类所需的根数作为零件的下料任务，原母板的宽度作为现在一维切割原料的长，这样又得到一个一维下料问题，同样借助一维下料模型的解法来获得局部近似最优解。经过上述两步后，二维下料问题就转化为了两个一维下料问题，在借助一维下料问题的求解算法得到两个局部最优解后，可以通过两者的结合得到最终解。

算法的基本思想是：

首先比较长的种类和宽的种类，从中选取种类比较少的一个作为第一次降维考虑的基础（在不影响一般性的前提下，以下假设宽度种类较少来进行描述）。按照宽度对所有零件进行分类，然后假设已经有各种宽度的模板足够多，而模板的长和原材料的长相等。这样在接下来的切割过程中将不考虑跨度问题，这样将完全变为一维下料问题。为了得到更优的解应该优先处理宽度最宽的一类，所以依据宽度给定每一类零件一个权值。同时要考虑到交货时间的要求，交货时间比较短的零件应该优先处理，所以依据交货时间给定每一类零件一个权值，两者的结合作为处理顺序的权值。在接下来的处理中，应该选取当前未处理集合中权值最大一类宽度的零件借助一维下料算法进行处理，以得到需要此类宽度模板的数量。为了提高原材料利用率，当一类宽度零件处理完毕后，如果有一些余料，将采用动态规划方法，在利用率高的要求下将其它宽度的零件尽量用这些余料来获得，直到剩下的余料不能再被使用为止。重复这个过程，可以得到每一类宽度的模板需要多少数目，同时得到一种下料方案。

接下来，将每一种宽度作为一维下料问题中零件的规格，而每一类宽度需要的数目就是一维下料问题中零件的数量要求，而此次一维下料问题的原材料长度是二维下料问题中原材料的宽度，对于这个一维下料问题借助上文的算法对其进行处理，得到一种下料方案。将第一次得到的一维下料方案和第二次得到的一维下料方案按顺序进行组合，即得到这个问题的下料方案。而第二次一维下料问题所需要的原材料个数就是在二维下料问题中所需要原材料块数。

算法描述:

比较所有零件每一维的类别数

对所有零件以其中类别数最少的一维为主进行排序

WHILE 能取出一个等长（宽）分组（按照权值顺序）

取出其中一个等长（宽）分组

在该分组内调用一维算法计算一组可行解

综合所有解，在另一维上得到一组满足第一问的原始数据

调用一维算法得到一维下料方案

将两次得到的下料方案进行组合，得到二维下料方案

IF 下料方案不能满足时间限制 THEN

调整权值重新计算，重新调用算法进行计算

4、结果与讨论

我们用 TC 编写了本文算法的程序，并将给定的实例代入程序进行了计算。对一维模型，经计算使用 800 根原料可以得到所有所需的零件，只超出最优量 3 根，废料总长度为 7232mm，共使用 58 种下料方式，原材料的利用率 $r=99.6\%$ 。对二维模型，经计算使用 451 块原料可以得到所有所需的零件，比最优用量超出 3 块，废料总面积为 7232mm^2 ，共使用了 66 种下料方式，材料利用率 $r=99.2\%$ 。图 1 给出其中一种下料方式

3	3	25		38	
		39	39	39	
3	3	25		38	
		39	39	39	

图 1 二维时给定实例的一种下料方式

从原材料的利用率，可以看出原材料得到较为充分的利用，这也说明模型和求解算法是十分有效和理想的。又因为算法设计时考虑了普遍的情况，所以算法在解决大多数实际下料问题，特别是大规模下料问题时是切实有效的。

5、模型应用

下料问题的本质是要在浪费最小的情况下，达到生产要求。在理论上，这类问题都可以用此模型求解。比如，我们对一维下料问题的求解可以推广到背包问题，对二维下料问题的求解可以推广到工具装箱问题。在解决二维下料问题中，我们设计了一种降维的方法，此方法具有通用性。因此对于任意维的下料问题、装箱问题或类似的其他问题都可以用本文中的模型和方法来处理。

相对于其它的非全局搜索算法，我们的算法具有以下优点：在规模较大的情况下，兼顾了计算速度和结果的质量，而且规模越大，算法得出的结果和目标结果越接近；相比其它剪枝的搜索方法，在时间允许的情况下，进行了自适应的调整，使结果尽可能满足可行性条件下的最优；相比直接在二维进行 DP 的算法，本算法的限制很少，对数据的要求很低。

参考文献

- [1] 左小德，梁云，薛声家. 线材的合理利用问题研究. 数学的实践与认识, Vol. 29 No. 3, 1999
- [2] 鲁习文. 玻璃划分的数学模型. 应用数学与计算数学学报, Vol. 13 No. 2, 1999
- [3] 宋翔，聂义勇. 无限制二维下料问题的改进动态规划算法. 信息与控制, Vol. 32 No. 1, 2003
- [4] 李春华. 线性规划在下料中的应用. 运筹与管理, No. 1, 1993