1.模型建立

|  |  |
| --- | --- |
| Sets |  |
|  | 表示seru即生产单元的符号， , C的索引 |
|  | 表示工人的符号，, 的索引 |
|  | 订单的批次, , 的索引 |
|  | 批次中所有产品的工序, , 的索引 |
|  | 批次b中产品的工序集合, , and  （一个批次假设b1中的产品工序集就是s1，批次b2中的产品工序集合就是s2，S包含s1 s2 s3……..） |
|  | 工人的工序集,  （一个工人会多种工序，可能全部都会，这样的叫全能工，可能只会产品工序的一部分，叫多能工。表示工人w1他会的工序集合） |
| Parameters 参数 |  |
|  | 批次的大小  （一个批次包含的产品数量，比如b1=30，就是批次1里面有30个相同的产品，这些产品相同，所要做的工序也就都一样） |
|  | 批次b中产品的工序s的标准处理时间。  （假设，b=1，s=1，就是，含义是批次b1中的这30个产品的某项工序s1，比如涂漆，标准处理时间是0.5h=） |
|  | 工人工序熟练程度。  （比如就是工人w1做涂漆工序s1的熟练程度为0.6，算是一般般熟练，工人熟练程度高，花完成工序时间就少，所以同样的工序要进行分配的目的在这里，分给做这个工序熟练程度高的工人，就有利于工人的工作量减少。） |
|  | 表示工人能否完成工序s。  （假设，=1就是工人w1可以做s1，否则=0。）  if equals 1 |
|  | 每个seru分配到的最大工人数。  （每个seru中不能有超过N个数的工人。） |
|  | 分配给每个工人的最多工序的数量。 |
| Variables决策变量 |  |
|  | 表示工人的分配。  （如果工人w1被分配到了c1中，那么，否则等于0）  if worker is assigned into *seru* , equals 1. |
|  | 批次b中被分到seru c的产品的量。  (假设结果显示，表明经过计算，批次b1中的15个产品被分到seru c1中。) |
|  | 如果批次b被分到了seru c中那么 equals 1.  （也是一个0-1变量，）表明b1批次中的部分产品或者全部产品被分到了seru c1中） |
|  | 表示工序的分配情况。  （0-1变量，=1表示的意思是批次b1中的产品工序s1被分给了在seru c3中的工人w2。这样工序的分配情况就出来了。如果等于0，那说明这个工序被分给了其他seru 中的某个工人，但是无论如何也要有工人去完成，这就有了下面的约束条件，保证每个工序都是有人要做的） |

|  |  |
| --- | --- |
| （这个式子是我的目标函数，min max就是我要最小化最大的工人工作量。  通俗理解就是，比如工人w2做了16个小时的工作，但是工人w3才做了4个小时，就存在着极度的不平衡，因此要将所有工人中做的最多的那个人的工作量最小化，直到分配结果显示，大家都是做8个小时，就很公平。只是举例子，结果不一定是这个数字。） | (1) |
| *s.t.*  约束条件 |  |
| 这一条保证了一个工人只能分给一个seru，不存在一个人同时分到两个seru中去，假设有3个seru，现在分配工人w1的去向，那么根据这个约束条件+，即他要不就分到了c1或者c2或者c3中。 | (2) |
| 这条保证在任意的一个seru中，所有的被分配到的工人不能超过seru能容纳的最大数量N。 | (3) |
| 这条的含义是在任意的批次b里，所有被分配到seru里面的产品加起来得和分配之前的总数是相同的。 | (4) |
| 对任意的工人、批次、工序集合、seru来讲， | (5) |
| 这条约束的意思是 对任意的工人、批次、seru来说，分给一个工人的工序总数不能超过这个人能承受的最大限度数量M。 | (6) |
| 这个是保证了一个工序只能分配给一个工人。 | (7) |
| 这条约束保证在seru中，比如seru c2中，分配到了工人和产品的情况下，也就是，=1，这样才可能有工序分到seru c2中，但也有可能不分配，即这种情况下的可能等于1，也可能是0。但是只要前面两个变量任意一个等于0，后面的工序就一定不会分配到这个seru里面。保证了工序分到seru中是要有人来完成的。 | （8） |
| 变量是个0-1变量 | (9) |
| 变量是个0-1变量 | (10) |
| 规定了批次被拆分后，分配到某个seru中的数量。当=0，比如 =0，说明批次b1被分到seru c2中的数量是0，但也有可能是10、9，总之不能拆过b1的原有产品的数量，最多是整体都在这个seru里面被生产了。 | (11) |
| ，  说明是个0-1变量 | (12) |
| 以批次b1和seru c3为例，如果=1，说明b1有产品被分到了c3去生产，那么由于这个条件的限制，（就是b1被分到c3的产品数量）最大也不会超过的量，但是如果=0，那么由于条件限制，就没有任何b1的产品被分到c3去生产。这个也是保证合理性，不存在没有分到seru中，却数量不为0。 | （13） |
| 因为是个0-1变量，要不就是0，或者1，但是它是个整数，可能是10、15等等，这样写的目的是如果那么也必然等于0，即批次中的产品如果分到某个seru中的数量是0，那么就是0。 | （14） |
| 这一条还是做个小假设，假设seru有三个，即C=3，那么对于批次b1来说，这一批的产品可能全部分给了一个seru里面，比如c2，也有可能分到了两个seru里，c1和c3，还有一种情况，就是三个seru都分到了b1里的产品，即它最少拆3次，也可能压根不拆，就直接整体被分到了某个seru里。但是拆的次数不可能拆过seru的数量。 | （15） |

The second objective termed as Total Labor Time can be expressed as

就是分到seru中某个批次的产品数量x工人的熟练程度x工人做这个工序的标准时间x工序是否有分到这个seru 的这个工人手上的成绩结果，在对所有c、b、s求和，就是上述是的含义。

。

描述一下问题：

有c个生产单元（seru），seru是日本的生产单元的叫法。b个订单批次，w个工人，每个产品都有s个工序。现在参数是已知的，见上述表格，需要求解的变量是四个，即哪些工人被分到哪些seru中去？批次拆分后，多少个产品被分到seru c1中，多少个被分到seru c2中？产品的哪些工序被分到哪个seru的哪个工人手里？生产分配的目的是，所有工人的工作量差不多，即每个人工作时间差不多，保证公平性。

混合整数非线性规划问题。需要用到benders分解将上述的这个数学模型分解成两个子模型，先求解一个子模型中出工人被分配到哪些sure中去的变量，再将其代入到另一个子模型中，求解剩下子模型的变量