

回溯算法在制丝生产线自动排产中的应用^①

刘 铝, 常炳国

(湖南大学 软件学院, 长沙 410082)

摘 要: 烟草企业面临着市场供给和计划生产的双重挑战, 烟草企业制丝生产线自动排产调度越来越受到重视, 优化制丝生产调度是提高生产效率的重要手段。论文分析了烟草制丝自动排产的业务流程和质量指标, 对制丝生产线的生产段、工艺柜等进行理论分析和科学建模, 采用回溯算法实现了制造生产调度自动排产与优化, 得到了预期效果。

关键词: 回溯算法; 制丝生产线; 自动排产

Application of Backtracking Algorithm to Automatic Arrangement in Tobacco Production Line

LIU Lv, CHANG Bing-Guo

(Software Institute, Hunan University, Changsha 410082, China)

Abstract: As the tobacco industry facing the dual challenges of the market and national plans, and automatic arrangement in tobacco production line becomes more and more important, optimizing production schedule becomes an important means to improve production efficiency. The paper analyzes the business process and quality indicators of automatically tobacco production, modeled production section, process cabinets, etc. in the tobacco production line, and in the end puts forward the solution based on backtracking algorithm.

Keywords: backtracking algorithm; tobacco production line; automatic scheduling

1 引言

卷烟企业生产线包括制丝生产线和卷包生产线两大部分, 其中制丝生产线对复烤后的烟叶叶片进行一系列的加工, 使其最终成为符合卷制要求的烟丝, 满足卷包生产卷制所需。该过程工序繁杂、工艺考究, 决定了卷烟产品的品质和产品风格的一致性和稳定性。如图 1 所示, 制丝生产线分为叶片、叶丝和加香三个生产段, 通过生产段后的烟丝经储叶柜、丝混柜和储丝柜, 工艺加工后满足卷包需求。制丝生产线按牌号生产。在制丝线生产过程中, 按批次进行加工, 相邻两个批次加工可以是相同牌号也可以不同牌号。实际生产中, 每个生产段在每个批次生产完成后, 由于中间存在卫生打扫等处理时间, 需要间隔一段时间才能进行下一个批次的生产, 不同牌号相邻批次间间隔时间比相同牌号相邻批次间间隔时间要长^[1]。制丝车间自动排产的总体目标是: 排出满足卷包进度计划

需求的制丝生产计划, 同时减少换牌时间(即相邻批次间的间隔时间)。实践中普遍采用的是经验判断和工艺实验相结合的方法: 批次调度人员对卷包需求进行经验判断, 制定批次调度计划, 通过工艺实验获得部分结果, 再进行调整, 如此反复, 直至满足卷包需求。这种经验判断的方式造成了人力、财力与时间的浪费, 而且难以达到换牌时间最少的优化效果。本文提出了一种应用回溯算法选择制丝生产最优排产工序, 并给出实验结果分析。

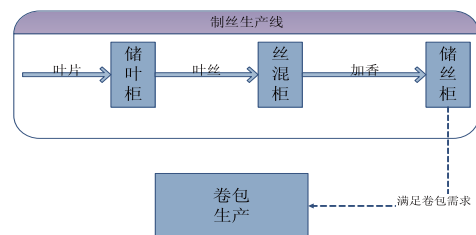


图 1 制丝生产线流程图

^① 收稿时间:2010-06-10;收到修改稿时间:2010-07-27

2 回溯算法

回溯法有通用解题法之称。用它可以系统地搜索问题的所有解。回溯法是一个既带有系统性又带有跳跃性的搜索算法^[2]。该方法在问题的解空间树中，按深度优先策略，从根结点出发搜索解空间树，算法搜索至解空间树的任一结点时，先判断该结点是否包含问题的解，如果肯定不包含，则跳过对以该结点为根的子树的搜索，逐层向其祖先结点回溯；否则，进入该子树，继续按深度优先策略搜索^[3,4]。回溯法寻找问题的所有解时，要回溯到根，且根结点的所有子树都被搜索遍才结束。这种以深度优先方式系统搜索问题解的算法称为回溯法，它适用于求解组合数较大的问题。

回溯法搜索解空间树时，通常采用两种策略避免无效搜索，以提高回溯法的搜索效率。其一是用约束函数在扩展结点处剪去不满足约束的子树；其二是用限界函数剪去得不到最优解的子树。这两类函数统称为剪枝函数^[5,6]。

回溯法相对于其他穷举的特点在于，不必把问题的每一层的所有可能解都遍历一遍，只要当前层的可能解不满足约束条件就抛弃该解，寻求下一个可能的解，而不必求解其余的下层解。如果当前层的所有可能解都不满足约束条件，则回溯到上一层，继续搜索^[7,8]。

3 实例分析

3.1 系统建模

根据烟草企业制丝生产线实际需求，可在叶片生产段前增加一个虚拟柜，假设工艺时间为 0，存储所有需要制丝的批次。假定 n 个批次需要进行生产，则在虚拟柜中 n 个批次可以按 $n!$ 个排列顺序进行生产。叶片生产段从虚拟柜中取制丝的批次，叶丝生产段从储叶柜中取制丝的批次，加香生产段从丝混柜中取制丝的批次。

将叶片、叶丝、加香三个生产段编号为 1、2、3 生产段，虚拟柜、储叶柜、丝混柜、储丝柜四个工艺柜编号为 1、2、3、4 号工艺柜。在不改变需求的前提下，可以假定不同批次在各生产段的间隔时间分为换批时间： T_batch 和换牌时间： T_brand ，各工艺柜需要的工艺时间： $T_cabinet_j (j = 1, 2, 3, 4)$ (其中虚拟柜工艺时间为 0, $T_cabinet_1 = 0$)，第 i 批次的需求完成时间： T_need_i ，出虚拟柜的开始时间： t_begin_i ，在第 j 号生产段的生产时间： $t_pro_{ij} (i = 1, 2, \dots, n, j = 1, 2, 3)$ ，进入第 j 号生产段时与第 $i-1$ 批次离开该生产段的间隔时间： $t_interval_{ij}$ (当第 i 批与 $i-1$ 批烟丝的牌号相同时，

$t_interval_{ij} = T_batch$ ，当第 i 批与 $i-1$ 批烟丝的牌号不同时 $t_interval_{ij} = T_batch + T_brand$)。

由此可得第 i 批次的完成时间：

$$t_complete_i = t_begin_i + \sum_{1 \leq j \leq 4} T_cabinet_j + \sum_{1 \leq j \leq 3} t_interval_{ij} + \sum_{1 \leq j \leq 3} t_pro_{ij}$$

生产过程中产生的间隔时间： $\sum_{1 \leq j \leq 3} t_interval_{ij}$ ，由此可知，必须满足 $t_complete_i \leq T_need_i$ ，同时使得

$$\sum_{1 \leq j \leq 3} t_interval_{ij} \text{ 最小。}$$

3.2 算法设计

给定个批次的集合，则问题的解空间可以组织成一棵树，从树的根结点到任一叶结点的路径定义了一个排产方案。图 2 是当 $n=3$ 时解空间的示例。其中从根结点 A 到叶结点 L 的路径上的标号组成一个排产的结果 P_1, P_2, P_3 。

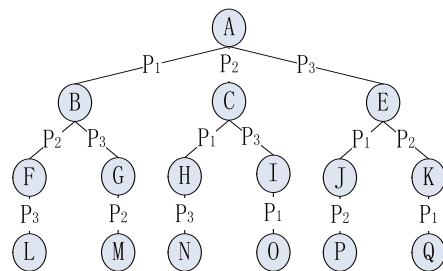


图 2 n=3 的解空间树

使用可行性约束函数剪去不满足约束条件 $t_complete_i \leq T_need_i$ 的子树。以结点 F 为例，当扩展至结点 F 时，以根结点 A 至结点 F 的顺序 P_1, P_2 判断 $t_complete_i \leq T_need_i (i = 1, 2)$ 是否全部满足，当存在 $i = 1, 2$ 使得 $t_complete_i \leq T_need_i$ 不成立，则以 F 为根的子树中不包含满足约束条件的解，因而该子树中的解均为不可行解，将该子树剪去，回溯至结点 F 的祖先结点 B。

使用限界函数剪去得不到最优解的子树。为构造最优解，在算法中记录当前最优解，即间隔时间 $\sum_{1 \leq j \leq 3} t_interval_{ij}$ 的最小值，在扩展结点处计算已产生的间隔时间，并与当前最优解的间隔时间比较，如果大于当前最优解间隔时间，则表示以该结点为根的子树中肯定不包含最优解，将子树剪去，回溯至当前扩展结点的祖先结点。当算法到达叶结点处，更新当前最优解。算法具体步骤如下：

步骤 1: 初始化当前最优调度方案，将其间隔时间

设为无穷大,从根结点开始以深度优先方式进行搜索;

步骤 2: 判断根结点至当前扩展结点的批次调度计划是否满足可行性约束函数 $t_complete_i \leq T_need_i$, 如不满足, 舍弃以当前扩展结点为根的子树, 转至步骤 5;

步骤 3: 计算当前扩展结点处已产生的间隔时间, 并与当前最优排产方案的间隔时间比较, 如大于当前最优排产方案的间隔时间, 舍弃以当前扩展结点为根的子树, 转至步骤 5;

步骤 4: 如果当前扩展结点为叶结点, 则将当前结点对应的排产方案更新为当前最优排产方案;

步骤 5: 如已无新的可扩展结点, 算法结束, 算法记录的当前最优解即为问题的最终解, 否则, 以深度优先的方式寻找新的扩展结点, 转至步骤 2。

为实现方便将限界函数值 $\sum_{1 \leq j \leq 3} t_interval_{ij}$ 最小, 转化为生产过程中的换牌次数最少。以下为部分核心源代码:

```
private int n; //作业数
private int interval; //当前调度所产生的间隔次数
private int best_interval = 1000; //最优调度的间隔次数
private int[] order; //当前调度
private int[] best_order; //最优调度
public void handle(int i) {
    if (i > this.n) { //叶结点, 解为当前最优解
        for (int j = 1; j <= n; j++) {
            this.best_order[j] = order[j];
            this.best_interval = this.interval;
        }
    } else {
        for (int j = i; j <= n; j++) {
            int flag = checkTime(order, j);
            //验证约束函数与限界函数,
            //满足, 继续向下遍历, 否则, 舍弃, 向上回溯
            if (flag != -1) {
                this.interval = flag;
                if (this.interval < this.best_interval) {
                    swap(order, i, j);
                    handle(i + 1);
                    swap(order, i, j);
                }
            }
        }
    }
}
```

3.3 效率分析

回溯法以深度优先的方式系统地搜索问题的解空间, 而问题的解空间是所有可能的决策组成。搜索过程中, 使用剪枝函数(约束函数和限界函数)避免无意义的搜索, 提高搜索效率。上述问题的解空间是所有批次的全排列, 因此算法在最坏情况下的复杂度为 $O(n!)$ 。

3.4 实验举例

以卷烟厂某天的批次计划为例, 生产开始时间为早上 8:00 开始, 储叶柜、丝混柜、储丝柜的工艺时间分别为 $T_cabinet_1 = 70$ 、 $T_cabinet_2 = 75$ 、 $T_cabinet_3 = 80$, 烟草分为三个品牌, 包括 A: 白沙; B: 精品白沙; C: 精品二代。各批次处理时间与需求时间如表 1 所示(单位: 分):

表 1 卷烟厂某天的批次计划

批次号	牌号	t_pro_{i1}	t_pro_{i2}	t_pro_{i3}	T_need_i
1	A	60	30	30	17:50:00
2	A	50	25	25	19:50:00
3	C	80	40	40	14:50:00
4	B	48	24	24	15:30:00
5	C	36	18	18	22:50:00
6	B	50	25	25	16:50:00
7	A	50	25	25	17:50:00
8	C	75	37.5	37.5	13:00:00

程序运行后得批次最优的排产方案, 批次顺序为 81273546, 所有批次均在需求时间内完成, 三个生产段中, 共有 9 次换牌处理。

4 结论与展望

本文将回溯算法应用于卷烟制丝生产线自动排产的问题中, 不仅解决了传统的依靠经验判断方式的一些缺陷, 达到了制丝生产车间对自动排产的总体目标, 满足卷包需求和减少换牌时间。通过卷烟工艺现场实验和应用证明, 该方法有效, 并具有一定的稳定性。

参考文献

- 1 王有国. 对烟草分组加工工艺技术的探讨. 科技信息, 2010,(3):352-355.
- 2 苏德福. 计算机算法设计与分析. 北京: 电子工业出版社, 2004.
- 3 周建军, 詹芹. 回溯法与分支限界法的用法取向探讨. 九江学院学报, 2009, 28(3): 18-21.
- 4 Priestley HA, Ward MP. A multipurpose backtracking algorithm. Journal of Symbolic Computation, 1994, 18(1): 1-40.
- 5 王发文, 马燕, 李宏达. 回溯法求解多约束分配问题. 江西师范大学学报(自然科学版), 2008, 32(6): 729-732.
- 6 王岚岚, 迟呈英. 不同数据结构下回溯法效率的比较. 沈阳师范大学学报(自然科学版), 2005, 23(4): 372-376.
- 7 Jiang H, Vidal JM. The message management asynchronous backtracking algorithm. Journal of Experimental & Theoretical Artificial Intelligence, 2008, 20(2): 95-110.
- 8 Stergiou K, Koubarakis M. Backtracking algorithms for disjunctions of temporal constraints. Artificial Intelligence, 2000, 120(1): 81-117.