

智能调度：离散制造行业车间智能排产调度方案

Contents

1 引言	1
1.1 概述	1
1.2 运筹相关模型介绍	1
2 总体蓝图	2
2.1 排程总体逻辑梳理	2
2.2 开发环境	2
3 APS 模型	3
3.1 日排产	3
3.2 精细排程	4
4 代码说明	6

1 引言

1.1 概述

制造业作为现代经济的重要组成部分，在全球范围内扮演着关键角色。在制造业中，生产调度是一个关键的环节，它对生产效率、产品质量和企业竞争力有着重要的影响。排产调度是指在有限的时间和资源的情况下，合理分配产品制造资源并确定工单排产加工的顺序，以完成特定的排产任务并实现预先设定的优化目标。随着市场需求的变化，消费者对个性化、差异化的产品需求越来越强烈，导致制造企业的生产订单越来越向小批量、多品种的方向发展，这种订单的趋势要求企业具备更高的生产计划和调度管理能力。然而，传统的车间排产调度方式通常是基于经验和规则的，缺乏灵活性和智能化，很难适应复杂的生产环境和动态的市场需求。因此，越来越多的制造企业开始迫切寻求智能化的排产调度解决方案，以提高生产效率、降低成本、提高产品质量和满足个性化需求。

1.2 运筹相关模型介绍

1 混合整数规划

混合整数规划 (Mixed Integer Program, MIP) 是 NP-hard 问题中的一类，它的目标是在线性约束下将线性目标最小化，同时使部分或全部变量均为整数值，在生产调度、资源分配与装箱等等现实场景中得到了广泛应用。

基本形式：

$$\begin{aligned} \min f(x) + g(y) \quad & s.t. \quad g(x) + h(y) \leq b \\ & x \in R_+ \\ & y \in N_+ \end{aligned}$$

求解方法：分支定界、割平面、拉格朗日松弛等。

2 约束规划

约束规划 (Constraint programming, CP) 的解决方案是一个值分配给每一个变量，使所有的约束满足。接下来，可以指定一个搜索方案来描述求解器如何从可能的分配中进行枚举，到实现一个解决方案。当搜索过程中出现矛盾时，约束规划有一个叫做回溯的功能来继续尝试其他可能的决定。约束规划在表达约束方面优于混合整数规划，因为它不局限于线性不等式。它采用约束传播推导机制，能够有效缩小问题可行域，适合解决如组合优化、排序问题等问题。

求解方法：约束传播、约束求解。

2 Local Search



约束规划 (Constraint programming, CP) 的解决方案是一个值分配给每一个变量，使所有的约束满足。接下来，可以指定一个搜索方案来描述求解器如何从可能的分配中进行枚举，到实现一个解决方案。当搜索过程中出现矛盾时，约束规划有一个叫做回溯的功能来继续尝试其他可能的决定。约束规划在表达约束方面优于混合整数规划，因为它不局限于线性不等式。它采用约束传播推导机制，能够有效缩小问题可行域，适合解决如组合优化、排序问题等问题。

求解方法：约束传播、约束求解。

2 总体蓝图

2.1 排程总体逻辑梳理

第一阶段：日排产，确定每个工单的生产日期，

- 决策变量
 - 每天每个设备生产哪些工单
 - 订单是否如期交付

- 优化目标：最大化按期交付率

第二阶段：精细排程，确定工作日每个设备上工单工作时间

- 决策变量
 - 各工单在设备的加工开始时间
 - 各工单所在的加工班次
 - 同一设备，B 工单是否紧接着 A 工单生产
 - 订单是否如期交付
- 优化目标：总切换时间最小

2.2 开发环境

编程语言	Python3
混合整数规划	scip
约束规划	Ortools SAT
启发式	local search



3 APS 模型

3.1 日排产

一、Model Formulation

Decision Variables $x_{wo,device,date}$:

$$x_{wo,device,date} \in \{0,1\}, \quad \forall (wo, device) \in wo_av_device, \quad date \in [start_day, end_day]$$

y_{order} :

$$y_{order} \geq 0, \quad \forall order \in sub_orders$$

$z_{prod,device,date}$:

$$z_{prod,device,date} \in \{0,1\}, \quad \forall prod \in sub_products, \quad device \in sub_devices, \quad date \in [start_day, end_day]$$

Objective Function

1、如期交付

$$\begin{aligned} \max \sum_{order \in sub_orders} & \left[10 \times 1_{\{start_day = \text{需求日期}_{order}\}} \right. \\ & \left. + (end_day + extend_day - \text{需求日期}_{order} + 1) \times 1_{\{start_day \neq \text{需求日期}_{order}\}} \right] \times y_{order} \end{aligned} \quad (3.1)$$

2、生产效率

$$\begin{aligned} \max \sum_{(wo,device,date) \in wo_device_date_select.keys()} & \text{需求量}_{wo} \times (2 \times 1_{\{\text{瓶颈物料号}_{wo} \\ & \in list_materials\}} + 1_{\{\text{瓶颈物料号}_{wo} \notin list_materials\}}) \times x_{wo,device,date} \end{aligned} \quad (3.2)$$

Constraints

1、设备产能时间约束

$$\begin{aligned} & \sum_{wo \in list_wo} \text{需求量}_{wo} \times \text{加工时间}_{wo,device} \times x_{wo,device,date} \\ & + \text{切换时间}_{device} \times \sum_{prod \in sub_products} z_{prod,device,date} \\ & + buffer_time \leq work_day_{device,date} \end{aligned}$$

2、物料使用量上限约束

$$\sum_{(wo,device) \in wo_av_device} x_{wo,device,date} \times \text{需求量}_{wo} \leq material_limit_{material,date}$$

3、唯一设备时间约束

$$\begin{aligned} & \text{Minimize } C_{\max} \\ & \sum_{\text{device} \in \text{sub_devices}} \sum_{\text{date} \in [\text{start_day}, \text{end_day}]} x_{\text{wo}, \text{device}, \text{date}} \leq 1 \\ & w_t \leq 2 \end{aligned}$$

4、工艺路径约束

$$\begin{aligned} & \text{Minimize } C_{\max} \\ & \sum_{\text{device} \in \text{sub_devices}} \sum_{\text{date} \in [\text{start_day}, \text{end_day}]} x_{\text{wo}, \text{device}, \text{date}} \leq 1 \\ & w_t \leq 2 \end{aligned}$$

5、前工序加工时，才能加工后工序

$$\begin{aligned} & \text{Minimize } C_{\max} \\ & \sum_{\text{device} \in \text{route_oper_av_device}_{\text{wo1}}} \sum_{\text{date1} \in [\text{start_day}, \text{end_day}]} x_{\text{wo1}, \text{device}, \text{date1}} \\ & \geq \sum_{\text{device} \in \text{route_oper_av_device}_{\text{wo2}}} \sum_{\text{date2} \in [\text{start_day}, \text{end_day}]} x_{\text{wo2}, \text{device}, \text{date2}} \end{aligned}$$

6、如期交付判定

$$y_{\text{order}} = \sum_{\text{device} \in \text{route_oper_av_device}_{\text{dict_order_tail_wo}[\text{order}]}} \sum_{\text{date} \in [\text{start_day}, \min(\text{total_day}, \text{需求日期}_{\text{order}} + 1)]} x_{\text{dict_order_tail_wo}[\text{order}], \text{device}, \text{date}}$$

7、产品是否加工识别

$$z_{\text{prod}, \text{device}, \text{date}} \geq x_{\text{wo}, \text{device}, \text{date}}$$

3.2 精细排程

一、约束规划

决策变量：

t_o : 工单 wo 开始加工时间

$x_{d,s,o}$: 二进制变量，表示工单 w 是否在设备 d 的班别 s 上加工

$y_{d,q1,q2}$: 二进制变量，表示在设备 d 上 o1 和 o2 是否存在相邻前后关系

$z_{d,s,o1,o2}$: 二进制变量，表示在设备 d 上，不同班次之间工单 o1 和 o2 是否相邻

目标：最小化切换总时间

$$\min \sum_{d,o_1,o_2} \text{switch_time}(d, \text{prod}(o_1), \text{prod}(o_2)) \cdot y_{d,o_1,o_2}$$

约束：

1. 工单开始结束时间约束

$$t_o \geq \text{shift_start}(d, s) \sum_d \sum_s x_{d,s,o} \quad \forall o \in O$$

$$t_o \leq \sum_{d,s} (\text{shift_end}(d, s) - \text{process_time}(o, d)) \cdot x_{d,s,o}, \quad \forall o \in O$$

$$\sum_{d,s} x_{d,s,o} \leq 1, \quad \forall o \in O$$

2. 工单唯一选择约束

$$\sum_d \sum_s x_{d,s,o} \leq 1 \quad \forall o \in O$$

3. 前后唯一工单约束

$$\begin{aligned} \sum_{o_2} y_{d,o,o_2} &= \sum_s x_{d,s,o}, \quad \forall d, \forall o \in O \\ \sum_{o_1} y_{d,o_1,o} &= \sum_s x_{d,s,o}, \quad \forall d, \forall o \in O \end{aligned}$$

4. 同班次时间间隔：前工序开始时间 + 加工时长 + 切换时长 ≤ 后工序开始时间

$$\begin{aligned} \sum_{o_2} y_{d,o_1,o_2} &= 1, \quad \forall d \in D \\ \sum_{o_1} y_{d,o_1,o_2} &= 1, \quad \forall d \in D \\ x_{d,0,o} &= 1, \quad \forall d \end{aligned}$$

5. 跨班次切换约束识别

$$t_{o_1} + \text{process_time}(o_1, d) + \text{switch_time}(d, \text{prod}(o_1), \text{prod}(o_2)) \leq t_{o_2} + M \cdot (1 - y_{d,o_1,o_2})$$

6. 跨班次切换时间约束

$$\begin{aligned} z_{d,s,wo_1,wo_2} &\geq y_{d,wo_1,wo_2} + x_{d,s-1,wo_1} + x_{d,s,wo_2} - 2 \\ t_{wo_2} &\geq \text{shift_start}(d, s) + \text{switch_time}(d, \text{prod}(wo_1), \text{prod}(wo_2)) + M \cdot (z_{d,s,wo_1,wo_2} - 1) \end{aligned}$$

6、工艺路径约束

$$t_{o_i} + \text{process_time}(o_i, d) \leq t_{o_{i+1}} + M \cdot (1 - x_{d,s,wo_{i+1}})$$

$$x_{d,s,o_i} \geq x_{d,s,o_{i+1}}$$

7、瓶颈物料约束

$$\sum_o x_{d,s,o} \cdot \text{material_demand}(o) \leq \text{material_limit}(\text{material}, \text{date})$$

二、启发式

Algorithm 1 启发式排程伪代码

- 1: **初始化和预处理**: day = start_day 到 end_day 没有班次信息 (day)
 - 2: **跳过当天的调度**
 - 3: work_orders \leftarrow 获取当天的工单 ()
 - 4: material_usage \leftarrow 初始化物料使用情况 ()
 - 5: order_work_orders \leftarrow 初始化订单工单列表 (work_orders)
 - 6: device_status \leftarrow 初始化设备状态 ()
 - 7: available_work_orders \leftarrow 筛选可用工单 (day, device_status)
 - 8: sorted_work_orders \leftarrow 排序工单 (available_work_orders, day)
 - 9: 排产工单 (sorted_work_orders, day, device_status, material_usage)
 - 10: 更新设备状态 (day, device_status)
 - 11: 更新订单完成时间 (day, order_work_orders)
 - 12: 更新物料使用情况 (day, material_usage)
 - 13: work_orders \leftarrow 移除已安排工单 (work_orders)
work_orders 为空
 - 14: **结束当天的调度**
 - 15: 记录未完成工单 (work_orders)
 - 16: 输出结果到 CSV 文件 (results)
-

4 代码说明

第三方库: numpy | pandas | ortools

运行 python aps_solver.py

修改数据文件路径:



```
if __name__ == '__main__':  
    path = r'决赛'  
    # 数据处理  
    product, line, work_day, material, day_num_map, switch_time, shift, zero_time \  
    = load_data(path)  
    # 结果文件名  
    file_name = '日排产计划_拆分'  
    # 日排程计算  
    daily_result = scip_model_stage1(product, line, work_day, material, day_num_map, \  
                                     switch_time)  
    # 精细排程计算  
    heuristic_stage2(product, line, material, day_num_map, switch_time,  
                    daily_result[['工单号', '日期', '设备']],  
                    shift, zero_time, f'{file_name}.csv')
```