

文章编号: 1001-0920(2013)05-0695-06

## 基于WLC的钢铁企业多工序、多路径订单投放方法

张春生, 李铁克

(北京科技大学 a. 东凌经济管理学院, b. 钢铁生产制造执行系统技术教育部工程研究中心, 北京 100083)

**摘要:** 针对按订单生产(MTO)下的多工序、多路径的钢铁生产中订单投放问题建立数学模型, 提出了度量订单交货提前/拖期程度、负荷均衡程度以及工序生产紧密衔接程度的3个指标. 针对问题的模型及其特点, 提出了基于负荷均衡的订单投放算法. 首先基于负荷控制的思想为订单池中的订单指定生产路径; 然后基于瓶颈工序优先均衡的原则调整工序负荷; 最后对非瓶颈工序进行负荷均衡. 基于企业实际生产背景的仿真实验结果验证了模型和算法的有效性.

**关键词:** 负荷控制; 多工序; 多路径; 按订单生产; 工艺路径选择

**中图分类号:** TP273

**文献标志码:** A

## Order release in steel enterprise with multi-production stage and multi-production route based on WLC

ZHANG Chun-sheng, LI Tie-ke

(a. Dongling School of Economics and Management, b. Engineering Research Center of MES Technology for Iron & Steel Production, Ministry of Education, University of Science and Technology Beijing, Beijing 100083, China. Correspondent: ZHANG Chun-sheng, E-mail: zhyutang@163.com)

**Abstract:** A mathematical model is established for order release in the make to order(MTO) steel enterprise with multi-production stage and multi-production route. Three indexes are built to measure the degree of delivery in advance or delay, the balance of workload, and the continuity of production in adjacent stage. According to the model and the characteristics of the problem, an order release algorithm based on workload balance is proposed. Firstly, the production routes are assigned for the orders in the order pool based on work load control(WLC). Then the workload of machines is coordinated based on the principle of the bottleneck stages. Finally, the load on non-bottleneck machines is balanced. The simulation experiments verify the effectiveness of the model and the algorithm.

**Key words:** workload control; multi-production stage; multi-production route; MTO; production route choose

### 0 引言

高附加值钢铁产品市场需求的扩大, 导致了钢铁产品的多样化和高端化, 生产方式逐渐由大批量、少品种转变为小批量、多品种, 钢铁生产流程呈现多工序、多工艺、长路径的特点, 生产组织更柔性, 物流关系更复杂, 生产管理的焦点在于负荷均衡、节奏一致以及生产连续. 某些企业为了适应市场需求为主导的形势, 采用了按订单生产(MTO)的经营模式. 在这样的生产环境下, 如何将生产订单投放到生产期间, 使得负荷均衡、物流连续、在制品库存减少、交货准时, 是钢铁企业面临的亟待解决的难题.

订单投放问题一直受到国内外学者的关注, 文献[1]将合同计划问题抽象成一种改进的旅行商问题, 用改进蚁群算法进行求解; 文献[2]针对钢铁企业的合同计划和余材匹配的集成优化问题建立模型, 提出一种嵌有“优先适合启发式”的遗传算法求解. 关于钢铁企业订单投放的现有研究多针对普钢的大批量生产, 总结出主要工序, 并假定订单经过相同的路径, 考虑因素多为合同的拖期和提前、库存的控制及余材的匹配, 以及设备能力的充分利用<sup>[3-5]</sup>.

钢铁生产中多工序、多路径的物流关系, 使得负荷、交货期与生产节奏三者之间的矛盾难以协调.

**收稿日期:** 2012-01-01; **修回日期:** 2012-06-09.

**基金项目:** 教育部博士学科点专项科研项目(20100006110006); 中央高校基本科研业务费专项资金项目(FRF-AS-09-007B); 国家自然科学基金项目(70771008).

**作者简介:** 张春生(1982-), 男, 博士生, 从事先进制造管理的研究; 李铁克(1958-), 男, 教授, 博士生导师, 从事先进制造管理、生产计划与调度等研究.

针对MTO模式的负荷控制(WLC)技术是一种利用输入/输出理论,通过控制各生产环节的负荷来均衡产能利用、缩短生产提前期及保证交货期的生产计划与控制方法<sup>[6-8]</sup>。订单投放是实现WLC的重要环节<sup>[9-10]</sup>。本文针对MTO下的多工序、多路径的钢铁生产中订单投放问题,基于WLC的思想,同时考虑负荷均衡、交货准时、生产连续等矛盾因素,提出基于负荷均衡的订单投放算法。

## 1 问题描述与模型

### 1.1 问题描述

钢铁企业的生产计划一般分为:能力与需求平衡、订单计划、批量计划、生产调度4个层次,对应的时间粒度分别为年/季、月、周、日。订单投放是第2层次需要解决的问题,目的是确定订单生产顺序,在计划周期内平衡产能,为进一步制定批量计划和生产日计划奠定了结构性的基础。该环节的流程为:首先计划系统从销售系统接收到客户订单,根据用户需求对订单进行库存匹配,若库存无法满足需求量,则根据订单要求进行工艺设计和质量设计,确定生产工艺路径,同时制定订单在各工序的物料需求数量、规格及质量标准<sup>[7]</sup>。本文将拆分到各工序的订单称之为工序订单,放入订单池。根据订单的需求量、交货期、工艺要求等属性,对工序订单的生产时间和机器的生产能力进行分配,最终形成确定了加工期间和加工机器的工序订单集合,为在周计划中进行组炉、组浇等批量计划提供条件。

钢铁产品品种、规格多样化,生产工艺也日益变得复杂,在复杂的工艺中炼钢部分包括连铸、模铸、电渣重熔等多条路径,加工部分有轧、锻、拔等多种加工方式,后续的产品处理包括酸洗、退火、矫直、抛丸、磨光等,使得生产组织变得柔性,也为负荷控制增加了复杂性。在处理多工序、多路径的订单投放时,难于兼顾工序生产节奏匹配与负荷在时间轴上的均衡,无法准确应答交货期,所以在生产过程中以一定的在制品库存协调工序间生产节奏的不一致,以保证产品的交货准时。除了考虑交货期因素以外,还要考虑负荷均衡、物流紧密,尤其是瓶颈工序负荷的均衡决定了整个生产系统的最大产能以及生产节奏的协调。订单工艺路径的选择决定着关键工序的定位,因而工艺路径的选择是订单投放环节进行负荷控制的重要因素。除此以外,订单投放问题涉及的约束还包括:1)生产工艺流程上各工序的能力约束,即产能约束;2)工序间特殊的等待时间限制,如连铸和热轧工序之间受限于一定的等待时间上限,热轧与冷轧之间受限于一定的等待时间下限;3)订单存在多条工艺路线,仅能指定一条路线。

本文建立的模型及算法基于以下几点假设:1)订单存在一条或多条已知工艺路线,每个工序对应一个机组;2)各工序对应的机组在一定时期内的加工能力一定并已知;3)订单在各个工序上的加工时间一定并已知;4)由于MTO模式下在制品库存受工序间生产节奏的影响,不作单独考虑;5)除了设备能力因素,其他生产资源的限制暂不被考虑;6)订单投放期间按5天(半旬)计算,在某工序的加工时间不跨期间<sup>[1]</sup>。

### 1.2 数学模型

#### 1.2.1 符号定义

为了便于描述问题,首先定义模型符号如下: $i$ 为生产订单序号,属于订单集合 $I$ ;订单 $i$ 的工艺路线集合定义为 $R_i$ , $r_i \in R_i$ ;  $J_{r_i}$ 为订单 $i$ 第 $r_i$ 条工艺路线上所有工序集合,其中第 $j$ 个工序表示为 $r_i(j)$ ,其对应的机组为 $m_{r_i(j)}$ ;  $m$ 为机组序号,属于机组集合 $M$ ;  $M_{r_i}$ 为订单 $i$ 第 $r_i$ 条工艺路线上所有机组集合;  $d_{ir_i(j)}$ ,  $p_{ir_i(j)}$ ,  $u_{ir_i(j)}$ ,  $l_{ir_i(j)}$ 分别为订单 $i$ 在其第 $r_i$ 条工艺路线上第 $j$ 个工序的物料需求量、所需加工时间、等待加工时间上限和下限;  $T$ 为计划周期,划分为多个生产期间 $t$ ;  $CT_{mt}$ 和 $CQ_{mt}$ 分别为 $t$ 期间内机组 $m$ 的可用加工时间及其最大产能;  $E_i$ 和 $L_i$ 分别为订单最早交货期和最迟交货期,用生产期间表示;  $\alpha_i$ 和 $\beta_i$ 分别为订单提前及拖期完成的惩罚系数;  $x_{ir_i(j)t}$ 为0-1变量,若 $t$ 期间订单 $i$ 在工艺路线 $r_i$ 的工序 $j$ 的机组上加工,其值为1,否则为0;  $y_{ir_i}$ 为0-1变量,若为订单 $i$ 指派的工艺路线 $r_i$ ,则其值为1,否则为0;  $t_{ir_i(j)}$ 为由订单 $i$ 和工序 $r_i(j)$ 表示的工序订单的投放期间。

#### 1.2.2 目标函数

针对多工序、多路径生产环境中交货期、负荷与生产连续3个矛盾因素,设计3个评价指标度量订单投放方案的优劣:订单完成提前/拖期惩罚、机组负荷在时间上的偏差程度及工序订单投放期间之间的距离,用公式表示如下:

$$\min f_1 = \sum_{i=1}^{N_I} (\alpha_i \times \max(0, E_i - t_{ir_i(N_J)})) + \sum_{i=1}^{N_I} (\beta_i \times \max(0, t_{ir_i(N_J)} - L_i)), \quad (1)$$

$$\min f_2 = \sum_{m=1}^{N_M} \sum_{t=1}^{N_T} \left| LD_{mt} - \frac{\sum_{t=1}^{N_T} LD_{mt}}{N_T} \right|, \quad (2)$$

$$\min f_3 = \sum_{i=1}^{N_I} \sum_{j=1}^{N_J-1} t_{ir_i(j+1)} - t_{ir_i(j)}. \quad (3)$$

其中

$$N_I = \text{card}(I), \quad N_J = \text{card}(J_{r_i}),$$

$$N_M = \text{card}(M), \quad N_T = \text{card}(T),$$

$$LD_{mt} = \sum_{i=1}^{N_I} d_{ir_i(j)} \times x_{ir_i(j)t}, m_{r_i(j)} = m.$$

### 1.2.3 约束条件

$$\sum_{r_i=1}^{\text{card}(R_i)} y_{ir_i} = 1, i \in I; \quad (4)$$

$$\sum_{t=1}^{\text{card}(T)} x_{ir_i(j)t} = 1, i \in I, r_i \in R_i, j \in J_{r_i}; \quad (5)$$

$$\sum_{i=1}^{\text{card}(I)} p_{ir_i(j)} \times x_{ir_i(j)t} \leq CT_{mt},$$

$$t \in T, m_{r_i(j)} = m \in M; \quad (6)$$

$$\sum_{i=1}^{\text{card}(I)} d_{ir_i(j)} \times x_{ir_i(j)t} \leq CT_{mt},$$

$$t \in T, m_{r_i(j)} = m \in M; \quad (7)$$

$$t_{ir_i(j)} = \{t \mid x_{ir_i(j)t} = 1, t \in T\}; \quad (8)$$

$$u_{ir_i(j)} \leq t_{ir_i(j+1)} - t_{ir_i(j)} \leq l_{ir_i(j+1)},$$

$$i \in I, r_i \in R_i, j \in J_{r_i}. \quad (9)$$

约束(4)表示任意订单指定的生产路径唯一;约束(5)表示每个工序订单指定一个生产期间,且仅能指定一个;约束(6)表示机组加工物料总时间不超过该机组在 $t$ 期间的可用时长;约束(7)表示 $t$ 期间机组的产能限制;约束(8)表示投放期间的取值约束;约束(9)表示相邻工序间物料加工等待时间上下限约束,因为 $u_{ir_i(j)}$ 和 $l_{ir_i(j)}$ 以生产期间表示,所以当等待时间上下限较小时,其取值为零。

## 2 求解算法

订单投放是负荷控制的核心,对于控制钢铁生产节奏、维持物流连续起着重要作用,其需要处理负荷均衡、订单准时交货和工序紧密衔接等矛盾因素.在复杂的生产物流关系网中,实现瓶颈工序的节奏控制和负荷均衡是控制整个生产系统平稳连续的关键,而订单工艺路径的变化可能导致瓶颈工序的改变,其成为算法设计过程中必须考虑的因素。

工艺路径固定的订单投放问题常常归结为TSP问题、多维约束的背包问题或VRP问题<sup>[1-2]</sup>,多路径的订单投放较之更为复杂.假设订单数为 $n$ ,若每个订单工艺路径数为 $r$ ,则订单生产方式的组合数为 $r^n$ ;若生产路径上有 $l$ 个工序,每个工序订单可以投放到 $t$ 个生产期间,则订单投放的组合数为 $t^{nl}$ .可见,可能的订单投放方案数随着订单数的增加呈指数级增长,这对算法的效率提出了要求。

基于以上分析,本文提出基于负荷平衡的订单投放算法WLB-OR(workload balance based order release algorithm),该算法分3步解决订单投放问题,并在此过

程中协调负荷、交货期和生产连续3个优化目标.首先基于负荷控制的思想为订单池中的订单指定生产路径;然后以瓶颈工序优先均衡的原则平衡机组负荷;最后对非瓶颈工序进行负荷均衡.以下进行详细描述。

### 2.1 指定生产路径

由于工艺路径的选择决定了物流网络的结构,首先需要解决的是工艺路径的确定.基于WLC思想,不考虑订单具体的投放期间,通过为订单指派工艺路径从整个计划周期的角度使工序间的负荷趋于均衡.本文用机组(或工序)的总负荷率 $RLD_m$ 来表示机组的产能利用情况,以机组负荷率集合 $\{RLD_m \mid m \in M\}$ 的方差 $V$ 来评价工序间的负荷均衡程度.具体步骤如下。

**Step 1:** 为每一订单随机指定一条可选的工艺路径。

**Step 2:** 根据工序订单的物料需求量 $d_{ir_i(j)}$ 计算各机组的负荷率 $RLD_m = \frac{\sum_{i=1}^{\text{card}(I)} d_{ir_i(j)}}{\sum_{t=1}^{\text{card}(T)} CQ_{mt}}$ ,  $m_{ir_i(j)=m}$ ;计算评价函数值 $V$ ,记最好的评价函数值 $V^* \leftarrow V$ ;设置已处理机组集合 $M^{\text{fix}}$ ,置空 $M^{\text{fix}}$ .令 $n$ 从 $1 \sim \max$ 执行Step 3~Step 5。

**Step 3:** 找出已处理机组集合以外的负荷率最大的机组(工序),作为待处理的机组,即 $m^{\text{unf}} \leftarrow \{m^* \mid RLD_{m^*} = \max(RLD_m), m \in M \setminus M^{\text{fix}}\}$ ,记此时的评价函数值为 $V^p$ ;如果找不出待处理的机组,返回Step 2。

**Step 4:** 确定安排到机组 $m^{\text{unf}}$ 上的订单集合 $I^* \leftarrow \{i \mid m_{ir_i(j)} = m^{\text{unf}}, i \in I\}$ ,令 $k$ 从 $1 \sim \text{card}(I^*)$ ,执行Step 5,最终如果 $V^* = V^p$ ,则 $M^{\text{fix}} \leftarrow M^{\text{fix}} \cup m^{\text{unf}}$ ,返回Step 3。

**Step 5:** 令 $i'$ 为订单集合 $I^*$ 的第 $k$ 个订单,且 $y_{i'r_i^*} = 1$ .令 $r_i'$ 从 $1 \sim \text{card}(R_{i'})$ 执行,若 $m^{\text{unf}} \notin M_{r_i'}$ ,则令 $y_{i'r_i'} = 1, y_{i'r_i^*} = 0$ ,计算此时的评价函数值 $V$ ,若 $V < V^*$ ,则 $V^* \leftarrow V$ ,返回Step 3;否则 $y_{i'r_i^*} = 1, y_{i'r_i'} = 0$ .最终如果 $y_{i'r_i^*} = 1, k \leftarrow k + 1$ ,返回Step 4。

### 2.2 初步指派生产期间

订单生产路径确定以后,工序订单得到了确定,以下将为工序订单指派生产期间.首先按工序订单最晚完成时间为其指派生产期间,则机组负荷集中在较晚的生产期间,通过将工序订单向较早的投放期间调整逐渐均衡机组负荷,同时避免了订单拖期问题.从实际应用角度考虑,瓶颈工序的资源利用是否合理直接关系到整个生产系统的产出和效率,所以订单投放中的负荷均衡策略为优先均衡瓶颈工序负荷.在平衡负荷的过程中兼顾了工序订单最早完成时间和工序

衔接的约束. 为了阐述方便, 引入机组期间的概念, 即某一生产期间的机组, 用符号  $W_{mt}$  表示, 其具有该阶段的最大产能、投放到上面的订单集合等属性.

对机组期间负荷的每一次调整都是一次工序订单移动的操作, 令  $o_{ir_i(j)}$  表示工序订单, 投放到  $W_{mt}$  上的工序订单集合为  $O_{mt} \leftarrow \{o_{ir_i(j)} \mid m_{ir_i(j)} = m \wedge x_{ir_i(j)t} = 1, i \in I\}$ . 工序订单的移动由以下步骤完成: 若工序订单  $o_{ir_i(j)}$  的投放期间减少一位, 即  $t_{ir_i(j)} \leftarrow t_{ir_i(j)} - 1$ , 则其所在机组期间  $W_{mt}$  上的工序订单集合  $O_{mt}$  移除此生产订单, 即  $O_{mt} \leftarrow O_{mt} \setminus \{o_{ir_i(j)}\}$ ; 该机组期间的前机组期间 (机组相同, 生产期间较之少一位) 增加该工序订单, 即  $O_{m,t-1} \leftarrow O_{m,t-1} \cup \{o_{ir_i(j)}\}$ ; 为了不违背约束 (10) 和 (11), 工序订单  $o_{ir_i(k)} (k \in J_{r_i} \setminus \{j\})$  也做同样的调整, 其相应的机组  $m_{r_i(k)}$  更新自身的负荷.

一个工序订单的移动导致多个工序订单移动, 以及多个机组负荷的变化, 所以调整机组期间的负荷时, 为确定移动哪个工序订单效果最好, 需要构造邻域结构. 移动一个该机组期间上的工序订单, 构成了一个邻域解. 工序订单在移动之前首先判断其是否达到最早完成时间, 如果已经是最早完成时间对应的期间, 则尽量不采取移动操作, 从而避免了订单的提前完成. 采用模型中的  $f_2$  作为评价函数来评价机组负荷在各生产期间的均衡程度, 选择评价函数值最小的邻域解, 即完成了一次机组期间负荷的调整. 工序订单生产期间的初步指派步骤如下:

Step 1: 计算工序订单最早、最晚完工时间

$$ect_{ir_i(j)} = E_i - \sum_{k=j+1}^{\text{card}(J_{r_i})} (p_{ir_i(k)} + s_{ir_i(k)}),$$

$$lct_{ir_i(j)} = L_i - \sum_{k=j+1}^{\text{card}(J_{r_i})} (p_{ir_i(k)} + s_{ir_i(k)})$$

其中  $s_{ir_i(k)}$  为订单在工序  $k$  的生产准备时间.

Step 2: 由最晚完工时间为工序订单指派生产期间  $t_{ir_i(j)} = \lceil lct_{ir_i(j)} / s^* \rceil$ ; 计算工序订单最早的生产期间  $et_{ir_i(j)} = \max(\lceil lct_{ir_i(j)} / s^* \rceil, 1)$ , 其中  $s^*$  为任一生产期间内的总生产时间.

Step 3: 计算机组期间的负荷率

$$\text{LDR}_{mt} = \frac{\sum_{i=1}^{\text{card}(I)} d_{ir_i(j)} \times x_{ir_i(j)}}{CQ_{mt}}, m_{ir_i(j)} = m;$$

找出最大负荷率的机组期间  $W_{m^*t^*}$ , 使得  $\text{LDR}_{m^*t^*} \leftarrow \max(\text{LDR}_{mt})$ , 若  $t^* = 1$ , 则转到 Step 8.

Step 4: 确定工序订单集合  $O_{m^*t^*} \leftarrow \{o_{ir_i(j)} \mid m_{r_i(j)} = m, x_{ir_i(j)t^*} = 1, j \in J_{r_i}, i \in I\}$ , 令  $h$  从  $1 \sim \text{card}(O_{m^*t^*})$  取第  $h$  个工序订单, 对其可否移动进

行检验, 转 Step 6; 若可以移动则执行 Step 7, 构造邻域解  $N(h)$ ; 若不能移动, 则令  $h \leftarrow h + 1$ .

Step 5: 选择使评价函数值最小的邻域解  $N(h^*)$ , 对工序订单  $o_{ir_i(j)} \leftarrow O_{m^*t^*}^*(h^*)$  执行 Step 7 的移动操作, 对相关机组期间的负荷进行更新, 转到 Step 3.

Step 6: 移动检验. 以  $-1$  为步长, 令  $k$  从  $j \sim 1$  执行, 若  $t_{ir_i(k)} - 1 - t_{ir_i(k-1)} < l_{ir_i(k-1)}$  且  $t_{ir_i(k)} = et_{ir_i(k)}$  ( $k \neq 1$ ) 时, 则工序订单  $o_{ir_i(j)}$  无法移动, 返回 Step 4; 以  $1$  为步长,  $k$  从  $j \sim \text{card}(J_{r_i})$  执行, 若  $t_{ir_i(k+1)} - (t_{ir_i(k)} - 1) > u_{ir_i(k)}$ , 且  $t_{ir_i(k+1)} = et_{ir_i(k+1)}$  ( $k \neq \text{card}(J_{r_i})$ ), 则工序订单  $o_{ir_i(j)}$  无法移动, 返回 Step 4.

Step 7: 移动操作. 以  $-1$  为步长, 令  $k$  从  $j \sim 1$  执行,  $t_{ir_i(k)} \leftarrow t_{ir_i(k)} - 1$ , 若此时  $t_{ir_i(k)} - t_{ir_i(k-1)} \geq l_{ir_i(k-1)}$ ,  $k \neq 1$ , 则结束此循环; 否则  $k \leftarrow k - 1$ . 以  $1$  为步长,  $k$  从  $j+1 \sim \text{card}(J_{r_i})$  执行, 若  $t_{ir_i(k)} - t_{ir_i(k-1)} \leq u_{ir_i(k-1)}$ , 则结束此循环; 否则  $t_{ir_i(k)} \leftarrow t_{ir_i(k)} + 1$ ,  $k \leftarrow k + 1$ . 更新相关机组期间的负荷, 返回调用步骤 Step 4 或 Step 5.

Step 8: 算法结束.

### 2.3 调整非瓶颈工序负荷

在工序订单生产期间的初步指派步骤中, 通过不断调整最大负荷率的机组期间上的工序订单, 机组上的负荷沿着期间从晚到早的方向均衡. 最先调整到第一个生产期间的机组为瓶颈工序对应的机组, 瓶颈工序对应的机组的负荷平衡以后, 需要对非瓶颈机组的负荷进行调整. 为了避免移动工序订单而影响到瓶颈工序的负荷, 在调整机组期间上某个计划加工的订单前, 对该订单进行检验. 工序订单如果已经到了最早完成时间对应的生产期间, 同样不优先进行移动. 具体步骤如下:

Step 1: 定义瓶颈工序机组集合  $M^k$ , 针对任一机组  $m$ , 若  $\text{LDR}_{m1} = \max \text{LDR}_{mt}$ , 则  $M^k \leftarrow M^k \cup \{m\}$ .

Step 2: 对机组期间集合  $W \leftarrow \{W_{mt} \mid m \in (M \setminus M^k), t \in T\}$ , 按机组负荷率递增顺序对其机组进行排序. 令  $k$  从  $1 \sim \text{card}(W)/2$  对第  $k$  个机组期间  $W_{m^*t^*}$  执行 Step 3, 对投放到上面的订单进行调整, 若对于所有  $1 \sim \text{card}(W)/2$  个机组期间都无法调整上面的订单, 则转 Step 6.

Step 3: 确定工序订单集合  $O_{m^*t^*} \leftarrow \{o_{ir_i(j)} \mid m_{r_i(j)} = m, x_{ir_i(j)t^*} = 1, j \in J_{r_i}, i \in I\}$ , 令  $h$  从  $1 \sim \text{card}(O_{m^*t^*})$  对第  $h$  个工序订单执行 Step 4, 检验其可否移动, 若无法移动则令  $h \leftarrow h + 1$ ; 若可以移动, 则执行 Step 5. 当工序订单发生了移动操作, 则返回 Step 1; 若任意  $o_{ir_i(j)} \in O_{m^*t^*}$  都无法进行移动操作, 则  $k \leftarrow k + 1$ , 返回 Step 2.

**Step 4: 移动检验.** 以-1为步长, 令  $s$  从  $j \sim 1$ , 若  $t_{ir_i(s)} - 1 - t_{ir_i(s-1)} < l_{ir_i(s-1)}$  且  $m_{r_i(s-1)} \in m^k$  或  $t_{ir_i(s)} = et_{ir_i(s)} (s \neq 1)$  时, 则工序订单  $o_{ir_i(j)}$  无法移动, 返回 Step 3; 以1为步长, 令  $s$  从  $j \sim \text{card}(J_{r_i})$ , 若  $t_{ir_i(s+1)} - (t_{ir_i(s)} - 1) > u_{ir_i(s)}$ , 且  $m_{r_i(s+1)} \in m^k$  或  $t_{ir_i(s+1)} = et_{ir_i(s+1)} (s \neq \text{card}(J_{r_i}))$ , 则工序订单  $o_{ir_i(j)}$  无法移动, 返回 Step 3.

**Step 5: 移动操作.** 与工序订单生产期间的初步指派步骤的 Step 7 方法相同.

**Step 6: 算法结束.**

## 2.4 算法复杂度分析

2.1 节的计算集中在 Step 3 ~ Step 5, 其为3个层次的循环操作, 最大可能执行次数为  $mnr$ , 在最内层循环每次调整订单生产路径后需要计算评价函数, 其计算时间为  $O(n)$ , 因此该步骤的复杂度为  $O(mn^2r)$ .

2.2 节的计算主要集中在 Step 3 ~ Step 5, 主要包括两层循环, 最大可能的执行次数为  $mtn \times n = mtn^2$ , 每次执行的操作都调用 Step 6 或 Step 7, 其计算时间为  $O(m)$ , 该步骤的计算复杂度为  $O(m^2n^2t)$ .

2.3 节中最外层循环代表工单的调整次数, 其最大可能数为  $mtn$ . Step 2 和 Step 3 对应第2、第3层循环, 最大可能执行次数仍为  $mtn$ . Step 4 和 Step 5 为 Step 3 调用, 其计算时间为  $O(m)$ , Step 2 中排序操作的计算时间为  $O(mt \log_2(mt))$ , 则该步骤计算复杂度为  $O(m^3t^2n^2) = \max\{O(m^2t^2n \log_2(mt)), O(m^3t^2n^2)\}$ .

综上所述, WLB-OR 算法的时间复杂度受制于 2.3 节的计算时间, 为  $O(m^3t^2n^2)$ .

## 3 计算实验

### 3.1 实验设计

为验证本文提出的订单投放模型与算法 WLB-OR, 以国内某钢铁企业多工序、多路径的生产特点为背景进行仿真实验. 实验分为2组: 第1组实验通过计算实际生产数据验证算法的有效性; 第2组实验随机生成实验数据, 针对不同的订单数量进行多组实验.

选择文献[11]提出的合同计划遗传算法(GEN)作为比较算法, 由于其针对相同路径的情况, 首先随机指定订单的工艺路径, 再按其方法对适合本文情况的相同数据进行实验. 主要评价指标为目标函数  $f_1 \sim f_3$  及计算时间.

### 3.2 实验数据及参数

订单的属性包括最早交货期、最晚交货期、产品需求量以及至少一条工艺路径, 如表1所示. 将计划月划分为6个生产期间, 每个生产期间设置为5天. 相同的生产设备称为机组, 每个机组的可用时间为  $24 \times 5 = 120\text{h}$ , 工序订单的物料需求量为  $d_{ir_i(j)} = d_i \times$

$\prod_{k=j}^{\text{card}(J_{r_i})} \delta_{ir_i(k)}$ , 其中  $d_i$  为订单需求量,  $\delta_{ir_i(k)}$  为工序耗钢系数. 工序间等待时间上下限  $u_{ir_i(j)}$  和  $l_{ir_i(j)}$  采用历史的平均值. 订单在机组上的生产时间为物料需求量除以机组加工速度, 即  $p_{ir_i(j)} = d_{ir_i(j)}/v_{im}$ ,  $m_{r_i(j)} = m$ , 其中  $v_{im}$  为机组  $m$  加工订单  $i$  的速度. 订单的提前及拖期完成的惩罚系数  $\alpha_i$  和  $\beta_i$  采用文献[11]的设置方法.

采用 C# 作为编程语言实现本文算法, 实验环境为 Pentium 4/ CPU 3.00 GHz/ RAM 1.50 GB 的台式机. 算法 WLB-OR 中指定生产路径的最大迭代次数设为 10; 遗传算法 GEN 最大迭代次数设为 300, 其余参数采用文献[11]的设置.

表1 部分订单工艺路径

订单	路径序号	工艺路径
order 1	1	电弧炉-精炼炉-模铸-棒材轧机-酸洗-退火
order 1	2	电弧炉-精炼炉-连铸-棒材轧机-酸洗-退火
order 2	1	电弧炉-精炼炉-连铸-退火-酸洗-带材轧机
order 3	1	电弧炉-精炼炉-连铸-线材轧机-精整
order 4	1	电弧炉-精炼炉-模铸-快锻机-精锻机-退火
order 5	1	电弧炉-精炼炉-电渣炉-快锻机-精锻机-退火
order 6	1	电弧炉-精炼炉-电渣炉-快锻机-退火-精锻机-退火
order 6	2	电弧炉-精炼炉-电渣炉-精锻机-退火-精锻机-退火
order 6	3	电弧炉-精炼炉-模铸-精锻机-退火-精锻机-退火
order 7	1	电弧炉-精炼炉-电渣炉-棒材轧机-退火-冷拔机-精整
order 8	1	电弧炉-精炼炉-电渣炉-棒材轧机-酸洗-精整
order 9	1	电弧炉-精炼炉-电渣炉-板材轧机-退火-酸洗-精整
order 10	1	电弧炉-精炼炉-电渣炉-线材轧机-退火-酸洗-精整
order 11	1	电弧炉-精炼炉-模铸-精整-棒材轧机-酸洗
order 11	2	电弧炉-精炼炉-连铸-精整-棒材轧机-酸洗

### 3.3 实验结果与分析

第1组实验中, 某钢铁企业某个月有323个订单参与订单投放, 分别用算法 WLB-OR 与 GEN 进行计算, 求解结果如表2所示.

表2 第1组实验计算结果

算法	目标函数			计算时间/s
	$f_1$	$f_2$	$f_3$	
GEN	757	6 271 772	736	2m26
WLB-OR	52	6 242 383	134	39

从机组的负荷率可看出, 退火工序为生产的瓶颈工序, 其在各生产期间的负荷分配如表3所示, 其中用  $v = \sum_{t=1}^{\text{card}(T)} \left| LD_{mt} - \frac{\sum_{t=1}^{\text{card}(T)} LD_{mt}}{\text{card}(T)} \right|$  测量各期间负荷与均值的偏差程度. 在第2组实验中随机生成订单最早、最晚交货期、产品需求量, 在工艺路径集中随机选取多条生产路径. 将订单根据不同的订单数量分为5组, 每组随机生成10个实例, 计算结果如表4所示.

表3 瓶颈工序负荷率

生产期间	机组负荷	
	WLB-OR	GEN
1	14 260.63	0
2	15 246.86	222.381 8
3	15 171.65	554.301 9
4	13 659.02	3 847.229
5	15 059.14	8 501.979
6	15 163.94	75 435.35
$v = 3\ 201.531$		$v = 121\ 350.3$

表4 第2组实验计算结果

订单数	WLB-OR			GEN		
	$f_1$	$f_2$	$f_3$	$f_1$	$f_2$	$f_3$
100	0	8 393 431	50	380	8 413 658	251
300	1	23 056 604	119	1 048	23 132 190	685
500	1	40 197 884	155	1 896	40 464 618	1 214
1 000	0	78 286 423	322	3 548	78 903 257	2 358
3 000	0	234 765 858	946	11 325	236 455 532	7 044

由以上实验结果可以得出以下结论:

1) 由于算法 WLB-OR 严格设定订单的投放期间范围, 比遗传算法将订单提前/拖期放入适应性函数进行优化具有更强的控制能力, WLB-OR 的提前/拖期惩罚值很小, 明显优于遗传算法, 这对于钢铁企业推进按订单生产的经营模式具有重要意义. 考虑到特殊钢铁产品的中间产品替代率较低, 算法 WLB-OR 控制订单生产在各工序的紧密衔接, 减少了库存, 所以目标函数  $f_3$  的值明显优于遗传算法.

2) 算法 GEN 将负荷超载的因素引入适应性函数中来调节机组的负荷, 与之相比, WLB-OR 在控制订单提前/拖期程度、订单工序紧密衔接的基础上均衡机组负荷, 因而从实验结果中可以看出算法 WLB-OR 的负荷均衡程度略好于算法 GEN; 算法 WLB-OR 优先均衡瓶颈工序的负荷, 所以瓶颈工序机组负荷的均衡程度明显优于算法 GEN.

3) 算法的计算时间随着订单数量增加而增加, 当订单数量达到 3 000 时, 求解时间为 1.2 h, 能够满足生产计划的实际需求.

综上所述, 基于负荷均衡的订单投放算法严格控制订单交货的提前和拖期, 缩短订单生产周期, 均衡设备的负荷, 尤其是对瓶颈工序负荷的均衡更具有较大优势, 从而充分利用了瓶颈工序的产能, 可见算法能够有效解决钢铁企业复杂生产工艺下的订单投放问题.

## 4 结 论

钢铁产品的市场结构正逐渐向高附加值的高端产品的方向调整, 钢铁生产流程不再是原有的固定

几个阶段, 而是向多工序、多路径的复杂流程转化. 生产管理过程中需要协调负荷、交货期及生产连续三者之间的矛盾. 基于负荷均衡的订单投放方法对于 MTO 模式的钢铁企业处理复杂工艺环境下的订单投放问题, 提升设备利用率、缩短生产周期具有实际意义. 本文对复杂工艺路径下的订单投放问题进行了分析, 建立数学模型, 提出了 3 个目标函数度量订单交货提前/拖期的程度、机组负荷各生产期间达到均衡的程度以及工序生产紧密衔接的程度. 在此基础上, 提出了基于负荷均衡的订单投放算法. 该算法分 3 个阶段逐步解决问题, 首先基于负荷控制的思想为订单池中的订单指定生产路径; 然后不断移动最大负荷率的机组期间上的工序订单, 通过瓶颈工序优先均衡带动其他工序负荷的调整; 最后对非瓶颈工序进行负荷的均衡. 数据实验表明, 模型和算法是可行和有效的. 在订单投放过程中将均衡负荷与其他智能搜索方法相结合, 利用智能方法的强搜索优势提高求解效率和质量是下一步的研究工作.

## 参考文献(References)

- [1] 张涛, 魏星, 张杰, 等. 基于改进蚁群算法的钢铁企业合同计划方法[J]. 系统管理学报, 2008, 17(4): 433-438.  
(Zhang T, Wei X, Zhang J, et al. Order planning method based on improved ant colony algorithm for the iron-steel plant[J]. J of Systems & Management, 2008, 17(4): 433-438.)
- [2] 卢克斌, 黄可为, 汪定伟, 等. 钢铁企业合同计划与余材匹配的集成优化方法[J]. 控制与决策, 2009, 24(1): 71-75.  
(Lu K B, Huang K W, Wang D W, et al. Integrated optimization approach of contract planning and surplus inventory matching of steel mill[J]. Control and Decision, 2009, 24(1): 71-75.)
- [3] 高聪, 唐立新, 唐建勋, 等. 禁忌搜索算法解决钢铁企业生产合同计划优化问题[J]. 东北大学学报: 自然科学版, 2009, 30(7): 944-947.  
(Gao C, Tang L X, Tang J X, et al. Tabu search for real order planning problem in steel industry[J]. J of Northeastern University: Natural Science, 2009, 30(7): 944-947.)
- [4] 阎艳, 王国新, 胡立臣, 等. 制造执行中订单投放顺序与工件加工批量集成决策[J]. 北京理工大学学报, 2008, 28(9): 773-777.  
(Yan Y, Wang G X, Hu L C, et al. Integrated decision making of order release sequence and lot sizing for manufacturing process execution[J]. Trans on Beijing Institute of Technology, 2008, 28(9): 773-777.)

(下转第715页)