

考虑轧辊成本的轧制计划编制与调整

董广静^{1,2}, 李铁克¹, 王柏林^{1†}, 栾治伟¹, 张春生³

(1. 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;
2. 首都师范大学 教育学院, 北京 100037; 3. 中国人民大学 信息资源管理学院, 北京 100872)

摘要: 在分析轧辊车削成本的基础上对轧制计划编制和调整问题进行深入研究, 建立以最小化轧辊车削量为目标的0-1整数规划模型. 针对问题特征, 构造基于替换邻域搜索的两阶段算法: 第1阶段, 对轧制单元实施最佳匹配和交叉使用调整; 第2阶段, 通过搜索轧制单元的轧辊替换邻域, 用车削量较少的轧辊替代车削量较大的轧辊, 从而优化第1阶段的有效解. 通过实际生产数据验证, 两阶段算法能够在满足轧制工艺规程的基础上有效降低轧辊车削量, 算法和模型均可行且有效.

关键词: 轧制计划; 车削成本; 替换邻域; 调整
中图分类号: TP29 **文献标志码:** A

Rolling planning and adjusting problem with roller cost

DONG Guang-jing^{1,2}, LI Tie-ke¹, WANG Bai-lin^{1†}, LUAN Zhi-wei¹, ZHANG Chun-sheng³

(1a. 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; 2. School of Education, Capital Normal University, Beijing 100037, China; 3. School of Information Resources Management, Renmin University of China, Beijing 100872, China)

Abstract: The problem of rolling planning and adjusting is considered. By analyzing the essential operation properties of the problem, a 0-1 integer programming model is established with the objective of minimizing the total cutting quantity of rolling plan. A two-stage algorithm based on exchange neighborhood searching is constructed. In the first stage, the initial solution is gained through matching and cross-using. In the second stage, the solution in the first stage is optimized by searching the roller neighborhood and selecting the alternative roller with less cutting quantity. The algorithm is verified by the actual production data, and the results show that the cutting quantity is reduced significantly, and the algorithm and model are feasible and effective.

Keywords: rolling plan; cutting cost; exchange neighborhood; adjustment

0 引言

轧制计划中钢坯的轧制顺序直接影响着成品的质量、工序能源消耗和生产成本等, 一个优秀的轧制计划不仅能够有效降低工序能耗, 而且能够大幅度降低轧辊车削成本、库存成本^[1]等. 轧辊是轧制生产过程中的一种非常重要的热工具, 不同钢产品是由不同孔型的轧辊轧制而成的. 在现实生产中, 轧辊的车削量是一项非常重要的成本项目, 特别是对于计划期内需要轧制的同一批钢坯, 往往会因为不同的轧制单元

顺序编制和交叉匹配使用方式, 导致轧辊在生产过程中的不同变动状态, 进而产生差异较大的车削成本. 因此, 有必要在满足轧制工艺的基础上, 找到一个最佳车削调度策略, 降低轧辊的车削成本. 一般在无缝钢管生产企业里, 生产部首先会确定一个粗轧制计划, 即某一时间段内需要轧制的轧制单元集合及对应轧制顺序, 且每个轧制单元所需的轧辊孔型都是已知确定的; 然后根据粗轧制计划到轧辊库内匹配与所需孔型相差最小的轧辊. 然而, 这种匹配并未考虑轧制单

收稿日期: 2015-09-13; 修回日期: 2016-01-28.

基金项目: 国家自然科学基金项目(71231001); 中央高校基本科研业务费专项资金项目(FRF-SD-12-012B, FRF-SD-12-011B).

作者简介: 董广静(1984—), 女, 博士生, 从事生产计划与调度、智能算法的研究; 李铁克(1958—), 男, 教授, 博士生导师, 从事生产调度与计划、先进制造管理等研究.

†通信作者. E-mail: wangbl@ustb.edu.cn

元之间的轧制顺序,无法有效地对轧辊进行重复利用,增加了轧辊车削成本,同时降低了生产效益.因此,有必要在编制轧制计划时引入轧辊库的实时轧辊信息,使其在轧制过程中降低车削成本.

在以往研究中,很少有文献在编制轧制计划时考虑降低轧辊的车削成本,研究主要集中在提高热装比、设备利用率以及优化生产调度过程^[2].文献[3]针对无缝钢管厂轧制计划编制问题做了深入研究,设计了基于邻域搜索的两阶段启发式算法;文献[4]建立了不确定轧制计划数的轧制计划VRP模型,构造了一种基于单亲遗传算子的免疫算法;文献[5]针对具有特殊生产工艺约束的带钢热轧计划编制问题,建立了基于多旅行商问题热轧轧制模型,并设计了新的Meta-heuristics算法;文献[6]将热轧批量计划编制问题归结为奖金收集的车辆路径问题,应用粒子群(PSO)算法进行求解,并对PSO算法的求解过程加以改进,使其适用于热轧批量计划问题;文献[7]针对炼钢-连铸中的机器冲突问题,给出了一种非线性规划模型,并将模型转化为线性规划模型求解;文献[8]提出了热轧计划调度问题的商业决策支持系统,并设计了一种具有生产特点的启发式算法;文献[9]研究了线圈轧制计划编制问题,设计了两阶段多项式启发式算法;文献[10]针对热轧调度问题设计了一种两阶段调度方法,并采用智能搜索算法调整轧制序列来提高热装率;文献[11]针对双目标TSP问题提出了多目标蚁群算法——Pareto最大最小蚂蚁算法,通过在6个标准测试函数上的实验及在热轧批量计划优化中的应用,验证了算法的优化性能和实用性;文献[12]针对热轧批量计划编制问题,建立了基于奖金收益车辆路径问题的计划数不确定的热轧批量计划模型,提出了一种基于模拟退火算法和蚁群算法结合的混合算法;文献[13]在分析轧钢一体化批量计划编制问题的基本特征基础上给出了一体化编制策略,建立了问题的约束满足优化模型,提出一种将改进离散粒子群算法、约束满足和邻域搜索相结合的混合算法.

传统的轧制计划编制方法在确定轧制单元顺序时的主要优化目标为切换轧辊次数,而具体采用的轧辊则是在确定顺序后再另行匹配,这种分步考虑方式对库存轧辊的利用和车削成本均为不利.对此,本文以某国有大型无缝钢管厂为研究背景,对冷装生产工艺下的轧制计划编制问题进行研究,建立以最小化轧辊车削量为目标的0-1整数规划模型,并构造基于邻域替换搜索的两阶段算法.首先,对轧制单元实施最佳匹配和交叉使用调整;然后,通过搜索轧制单元的轧辊替换邻域,用车削量较少的轧辊替代车削量较大的轧辊,优化第一阶段有效解.

1 问题描述与分析

一个完整的轧制计划是由多个轧制单元按照一定的轧制顺序组成,每个轧制单元是由一种规格的生产合同组批而成,在组批的过程中主要考虑合同要求,如交货期、规格、钢种等因素,每轧制完一个轧制单元需要切换轧辊,使其适用于下一个轧制单元.不同规格的钢管成品是由不同孔型的轧辊轧制而成,而所需轧辊若无法在轧辊库中找到相同孔型的库存轧辊,则需将孔径较大的库存轧辊通过车削工艺改为所需轧辊的孔型,进而产生车削成本.在热轧生产中,这种情况是相当普遍的,往往轧辊的每次使用都需要车削,即轧辊从当前轧制单元使用完毕后到下一个轧制单元使用之前必须经过车削这一道工序.车削操作是按整体进行,而且是一次性消耗工具,即重复着“使用-车削-再使用”的过程,直到轧辊报废.由于在实际生产中,车削成本较为昂贵,且车削过程会影响到生产连续性,因而在实施轧制计划时要求尽可能减少轧辊车削量.

在实际的生产管理中,当粗轧制计划由生产部下发后,即确定了计划期间内需要轧制的轧制单元集合及轧制顺序,以及各轧制单元所需的轧辊孔型(轧辊规格).此时,如何为每个轧制单元匹配最佳的车削轧辊,如何在满足轧制工艺的前提下调整轧制单元的轧制顺序,这两个问题的合理解决对控制轧辊车削量具有重要影响.但现有的研究往往是关注单个轧制单元的轧辊匹配策略,而忽略了后一个问题,即从整个轧制计划的角度来考量轧制顺序对于车削成本的影响.图1的示例说明了轧辊顺序对于轧辊车削量的影响.其中,初始轧制计划(轧制单元序列)为 $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$,所需轧辊规格分别为 $\{120, 115, 110, 121, 127\}$;轧辊库内有5根轧辊,规格分别为 $\{117, 119, 128, 132, 135\}$,并根据轧辊规格递增顺序进行编号(图1中“/”前为轧辊编号,其后为规格参数).方案1为依次为轧制单元匹配最佳车削轧辊的结果,车削量为39个单位.经分析发现,若对轧制单元的轧制顺序进行调整,同时考虑轧辊的交叉使用,则能够有效减少轧辊的车削量和使用量,即给出的方案2.在方案2中,通过将轧制顺序调整为 $5 \rightarrow 2 \rightarrow 4 \rightarrow 1 \rightarrow 3$,在轧辊匹配过程中仅使用了编号为1、3、4的3根轧辊,车削量便降低了26个单位.

可以看出,在满足轧制生产工艺和轧辊切换次数要求的前提下,同时考虑轧制单元顺序和轧辊重复匹配来实施调整策略,能够显著降低轧制计划的总车削成本.本文将基于这一思路对问题的模型和算法展开研究.

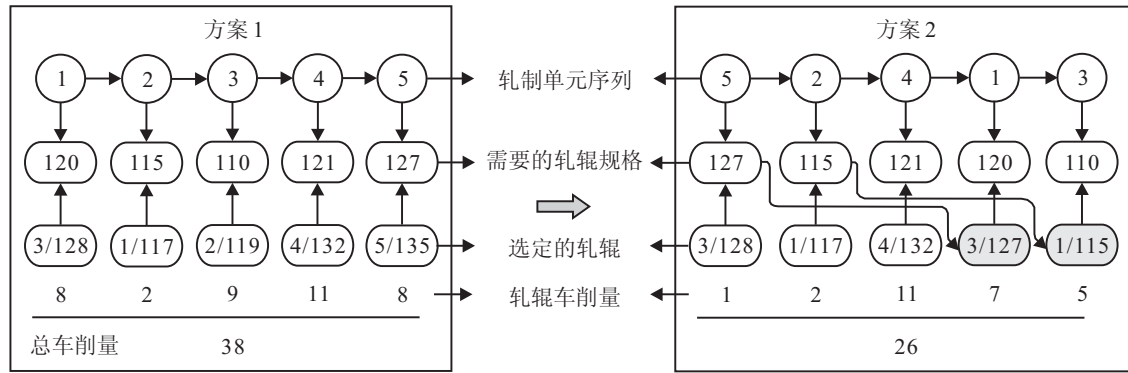


图 1 轧制单元的轧制顺序对轧辊车削的影响

2 问题建模

2.1 模型假设

- 1) 初始轧制计划是已知确定的, 其中轧制单元集合及其之间的轧制顺序是给定的.
- 2) 轧制单元之间的切换轧辊次数已知, 而且只有轧制规格差在一定范围内时才允许切换, 以避免轧制规格之间的大幅度跳跃.
- 3) 每个轧制单元对应一种孔型的轧辊.
- 4) 虽然在生产过程中, 与轧辊车削成本相关的因素有很多, 如人力成本、车削设备消耗成本、吊车运行成本等, 然而这些成本在钢厂的运行过程中均被视为资产的消耗成本, 即固定的消耗备件, 因而在计算机车削成本时对上述因素不予考虑.

2.2 符号定义

为了便于描述模型和理解算法, 各符号定义如下.

- 1) 索引及集合.
 - i : 轧辊编号, $i \in I$, I 为轧辊集合;
 - j : 轧制单元编号, $j \in J = \{1, 2, \dots, |J|\}$, j 为轧制单元的初始轧制顺序, J 为轧制单元集合.
- 2) 模型参数.
 - s_j : 轧制单元 j 所需的轧辊孔型;
 - h_i : 轧辊库内轧辊 i 对应的孔型;
 - g_{ij} : 轧制单元 j 选择轧辊 i 车削的轧辊孔型, $g_{ij} > s_j$;
 - S_j : 轧制单元 j 对应的候选轧辊集合;
 - $c_{jj'}$: 轧制单元 j 切换到轧制单元 j' 时轧制的切换轧辊数;
 - U_j : 轧制单元 j 对应的轧制位置;
 - C : 轧制单元之间切换轧辊的最大切换数.
- 3) 决策变量.
 - x_{ij} : 等于 1 时, 轧制单元 j 选择轧辊 i 进行轧制, 否则为 0;

$z_{jj'}$: 等于 1 时, 轧制单元 j 与轧制单元 j' 交换轧制位置调整后交叉使用, 否则为 0;

$l_{jj'}$: 等于 1 时, 轧制单元 j 与轧制单元 j' 直接交叉使用, 否则为 0.

2.3 问题模型

以最小化轧辊车削量为目标的轧制计划模型建立如下:

$$\min F = \sum_{i \in I} \sum_{j \in J} (g_{ij} - s_j) x_{ij}. \quad (1)$$

$$\text{s.t.} \quad \sum_{i \in I} x_{ij} = 1, \quad j \in J; \quad (2)$$

$$\sum_{j \in J} x_{ij} \leq 1, \quad i \in I; \quad (3)$$

$$c_{jj'} y_{jj'} \leq C; \quad (4)$$

$$s_{j'} = g_{ij}, \quad U_j \Leftrightarrow U_{j'}, \quad \text{if } z_{jj'} = 1; \quad (5)$$

$$g_{i'j'} = s_j, \quad \text{if } l_{jj'} = 1; \quad (6)$$

$$g_{ij} x_{ij} \in S_j = \{i | s_{j'} > g_{ij'} x_{ij'}, \forall U_{j'} < U_j\}; \quad (7)$$

$$x_{ij} = 1, \quad i \in S_j, \quad j \in J; \quad (8)$$

$$x_{ij}, z_{jj'}, l_{jj'} \in \{0, 1\}. \quad (9)$$

可以看出, 此模型为 0-1 整数规划模型. 其中: 目标函数 (1) 表示最小化轧制计划实施过程中轧辊的车削总量; 约束 (2) 表示每个轧制单元匹配一个车削轧辊, 且只能匹配一个; 约束 (3) 表示每个库内的轧辊的两种状态: “被选中”、“未被选中”; 约束 (4) 表示轧制单元之间切换轧辊的次数不能超过给定上限值, 以避免切换轧辊数过大, 造成吊车和切换时间成本过高; 约束 (5) 表示轧制单元之间轧制位置调整后交叉使用时, 车削轧辊的交叉使用和轧制位置赋值过程; 约束 (6) 表示轧制单元之间车削轧辊交叉使用赋值过程; 约束 (7) 表示轧制单元所选的轧辊来自其候选集合内, 其候选集合中包含了所有在它之前轧制的下线轧辊, 该式随轧制单元的匹配调整而实时变化; 约束 (8) 表示每个轧制单元必须在其候选轧辊集合中选取车削轧辊; 约束 (9) 定义了决策变量的取值范围.

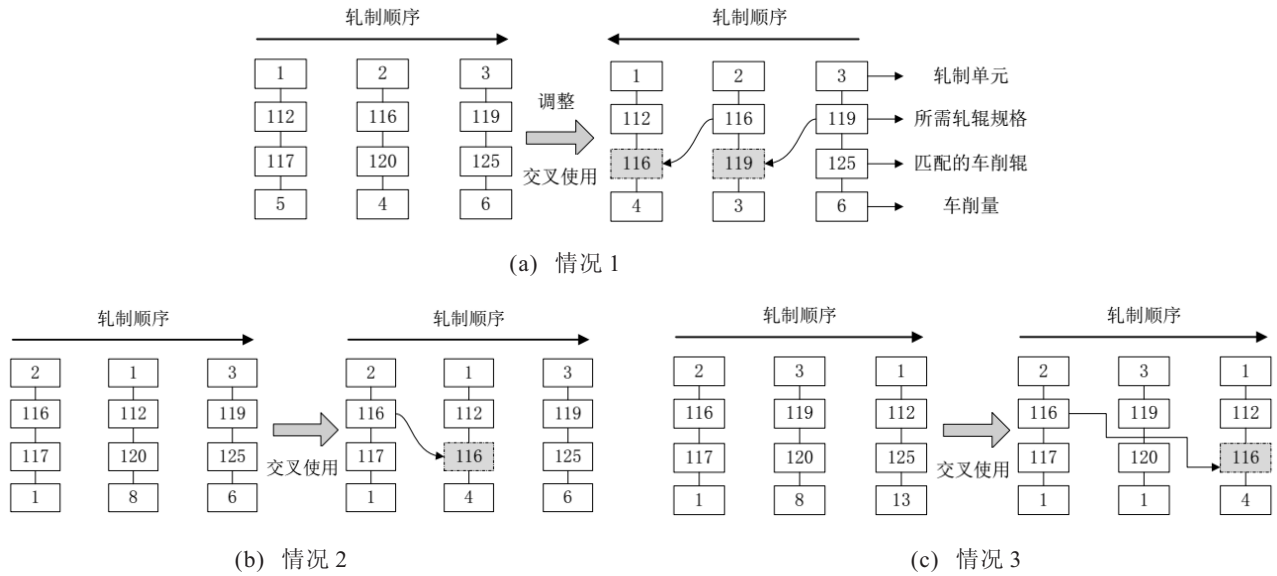


图 3 轧制顺序调整和交叉使用的几种算例

Step 3: 计算每个轧制单元的最小车削量值 $u_k = \min\{u_{ik}\}, \forall i \in U_k, k \in J$;

Step 4: 将轧制单元按照最小车削量值非减排序;

Step 5: $k = 1$, 对应第 1 个轧制单元;

Step 6: 在对应替换邻域中选择车削量最少的轧辊 i_k 匹配, 有 $J = J - \{i\}, I = I - \{i_k\}$;

Step 7: 若 $k > |J|$, 则结束匹配, 输出结果, 否则 $k = k + 1$, 转 Step 2.

第 2 阶段: 交叉使用调整 (CA).

Step 1: 初始化第 1 阶段结果, 并将轧制单元按照最初的轧制顺序逆序排序;

Step 2: $k = 1$, 对应第 1 个轧制单元;

Step 3: 计算轧制单元 k 的可交叉匹配使用轧制单元集合 A_k , 若 $A_k = \emptyset$, 则转 Step 4, 否则转 Step 5;

Step 4: 计算轧制单元 k 与集合 A_k 中各轧制单元之间的车削量, 选择交叉使用后, 车削量降低幅度最大的轧制单元作为交叉使用目标, 建立交叉使用关系;

Step 5: 若 $k > |J|$, 则结束交叉使用匹配, 输出结果, 否则 $k = k + 1$, 转 Step 3.

3.2 基于替换邻域的轧制单元顺序调整方法

在执行轧制计划的过程中, 轧制单元之间的车削轧辊在满足轧制时间约束的情况下是可以继续提供给后续的轧制单元的, 本节将针对这一特点对轧制计划实施调整. 由图 3 的示例可以看出, 调整轧制单元 1 和 5 的轧制顺序, 对应的车削量从以前的 16 个单位降低到 8 个单位, 优化效果非常可观. 考虑到每个轧制单元具有多个候选轧辊, 轧制单元之间不仅存在轧辊匹配优化, 也存在轧辊交叉使用优化, 各因素彼此之间相互制约, 因此, 本文设计了基于替换邻域搜

索轧制单元顺序调整算法 (A-ANS). 由于在轧辊库存量较大时, 轧制单元对应的替换邻域范围会很大, 这里设置替换邻域是在一定范围内的邻域, 可以表示为 $U_j = \{h_i > s_j, i \in I\}, \|U_j\| \leq 10$, 其中 $\|\cdot\|$ 表示集合的元素个数. A-ANS 算法的主要思想是: 在第 3.1 节初始解的基础上, 替换轧制单元的轧辊替换邻域内的车削轧辊, 同时调整轧制单元之间的轧制顺序来改善第 1 阶段的有效解. 具体步骤如下.

Step 1: $k = 1$, 对应第 1 个轧制单元.

Step 2: 判断轧制单元 k 是否已被交叉使用, 若被交叉使用, 则转 Step 6; 否则转 Step 3.

Step 3: 计算轧制单元 k 对应的可交叉使用的轧制单元集合 $P = \{j | h_k < s_j < g_{ij}\}$, 若 $P = \emptyset$, 则转 Step 6; 否则转 Step 4.

Step 4: 令 c 为可交叉使用的位置惩罚项, 若可交叉使用的轧制单元在 k 之前轧制, 则 $c = 2$; 否则 $c = 0$. 将集合 P 按照 $g_{ij} - s_k + c$ 值非减排序, 锁定第 1 个轧制单元 P_1 , 若 $c = 0$, 则直接标记交叉使用; 若 $c = 2$, 则替换轧制单元, 标记交叉使用.

Step 5: 检验序列是否满足约束 (6), 若满足, 则执行交换, 转 Step 6; 否则令 $P = P - P_1$, 转 Step 4.

Step 6: 若 $k > |J|$, 则结束替换调整; 否则令 $k = k + 1$, 转 Step 2.

3.3 TSAANS 算法

TSAANS 算法集成了 BM-CA 算法与 A-ANS 算法的优化过程, 主要分为两个阶段: 第 1 阶段通过 BM-CA 算法为每个轧制单元匹配合适的车削轧辊, 同时当前匹配的基础上实施轧辊的交叉使用; 第 2 阶段通过搜索轧制单元的轧辊替换邻域, 用车削量较少的轧辊替代车削量较大的轧辊, 实施替换调整策略以优

化第1阶段有效解. 具体算法求解步骤如下.

Step 1: 初始化 BM-CA 算法的初始解 T , 最优解为 T^* , 令 $T^* = T$.

Step 2: $k = 1$, 对应第 1 个轧制单元.

Step 3: 构造轧制单元 k 对应的替换邻域 U_k .

Step 4: $k_1 = 1$, 对应 U_k 中第 1 个交换轧辊.

Step 5: 将轧制单元的车削轧辊替换为轧辊 k_i , 调用 A-ANS 算法对轧制计划实施调整策略, 计算当前对应的有效解 T' . 若 $T \leq T'$, 则令 $T = T'$; 否则令 $T = T'$, 转 Step 6.

Step 6: 若 $k_i > |U_k|$, 则转 Step 7; 否则 $k_i = k_i + 1$, 转 Step 5.

Step 7: 若 $k > |J|$, 则结束替换调整, 输出最优解; 否则令 $k = k + 1$, 转 Step 3.

上述 TSAANS 算法实现了在考虑轧辊车削成本前提下对轧制顺序的调整, 是在第 2.1 节优化匹配 BM-CA 算法基础上对轧制计划的进一步优化. 由于轧制计划编制问题是一个 NP 难问题, TSAANS 算法通过调整过程中解的比较迭代得到实际生产可用的有效解, 能够在满足生产工艺要求下降低轧辊车削成本, 提高生产效率.

3.4 算法复杂度分析

设轧制单元数为 n , 轧辊库内轧辊的个数为 m , 每个轧制单元对应的候选轧辊个数为 n_i ($i = 1, 2, \dots, n$).

BM 算法外部循环 $k = 1 \rightarrow n$, 内部循环 $l = 1 \rightarrow m$, 计算复杂度为 $O(nm)$; 在 Step 2 中调用排序算法, 计算复杂度为 $O(n \log_2(n))$; 在 Step 5 ~ Step 7 中依次为轧制单元匹配最佳轧辊, $k = 1 \rightarrow n, l = 1 \rightarrow n$, 计算复杂度为 $O(n^2)$, 因此, BM 算法计算复杂度为 $O(nm + n^2)$. CA 算法主要实现为 BM 算法输出的结果进行交叉使用计算, $k = 1 \rightarrow n, l = 1 \rightarrow n$, 计算复杂度最差为 $O(n^2)$. 因此, BM-CA 算法的计算复杂度为 $O(nm + n^2)$.

TSAANS 算法外部循环 $k = 1 \rightarrow n$, 对每个轧制单元的替换邻域集合 U_k 中变量遍历调用 A-ANS 算法实施调整策略, $i = 1 \rightarrow n$, 其中 Step 2 计算可交叉使用轧制单元集合 $k = 1 \rightarrow n$, Step 3 中对集合 P 进行非减排序 $j = 1 \rightarrow n$, 所以 TSAANS 算法的求解复杂度为 $O(n^2 \log_2(n))$.

综合上述分析, 本文设计的两阶段启发式算法的求解复杂度为 $O(n^2 \log_2(n) + mn)$.

4 仿真实验

4.1 实验数据与环境

为了验证 TSAANS 算法的有效性, 从某钢厂的

实际生产数据中抽取出一个含有 100 个轧制单元的轧制计划实施编制. 在实验过程中, 从 100 个轧制单元中随机抽取 10、15、20、 \dots 、100 个轧制单元计划, 分为 19 组数据实验, 重复实验 30 次. 算法采用 Microsoft Visual Studio C# 编程实现, 测试环境为 Pentium 4/1 GHz/2 G/Windows XP Professional.

实验主要是将文中设计的 TSAANS 算法的两个阶段与改进的人工求解方法进行对比. 传统人工求解的主要思路是: 为每个轧制位置匹配切换机架次数最少的轧制单元, 依次为轧制单元匹配车削量最少的轧辊, 不考虑轧辊的交叉匹配. BM 算法是在传统人工求解方法的基础上, 按照替换邻域内轧辊与轧制单元所需轧辊差值的逆序依次为轧制单元匹配最佳轧辊的过程, 这里称为“改进的人工算法”.

实验 1 固定轧辊库存信息和库存轧辊孔型种类, 在轧制单元集合数变化的情况下, 将 TSAANS 算法、BM-CA 算法和 BM 算法在优化轧辊车削量上进行对比.

实验 2 固定计划中轧制单元数, 增大轧辊库内轧辊数量和轧辊孔型种类, 对上述 3 种求解算法的车削量的变化情况进行对比.

4.2 实验结果与分析

实验 1 主要验证固定轧辊库存信息和轧辊孔型种类, 改变轧制单元数时, 3 个算法的改进程度, 这里固定轧辊库内的轧辊数量和轧辊孔型种类均为 200.

实验 2 主要是验证改变轧辊库内轧辊数量及种类, 固定轧制单元长度时, 3 个算法的有效性, 这里固定轧制单元长度为 60. 实验结果如表 1 和表 2 所示. 其中: m 为轧制单元数, n 为轧辊库内轧辊的数量和种类数, R_g 、 B_c 、 T_s 分别为改进人工求解算法、BM-CA 算法、TSAANS 算法的求解结果, p_1 、 p_2 分别为 TSAANS 算法相对 BM-CA 算法和 BM 算法结果的提高比率.

表 1 实验 1 的计算结果

m	R_g	B_c	T_s	$p_1/\%$	$p_2/\%$
10	12.838	12.597	12.361	1.873 5	3.715 5
20	25.877	25.57	25.259	1.216 3	2.388 2
30	42.797	41.247	39.775	3.568 7	7.061 2
40	56.492	54.063	52.006	3.804 8	7.940 9
50	73.588	68.873	65.083	5.502 9	11.557 6
60	88.832	85.412	78.838	7.696 8	11.250 5
70	108.018	98.992	90.994	8.079 4	15.760 3
80	142.912	120.05	107.532	10.427 3	24.756 5
90	160.914	136.244	116.606	14.413 8	27.535 2
100	184.804	153.88	135.264	12.097 7	26.806 8

表 2 实验 2 的计算结果

n	R_g	B_c	T_s	$p_1/\%$	$p_2/\%$
70*70	89.638	74.056	66.35	10.405 6	25.980 1
80*80	85.748	71.906	64.216	10.694 5	25.110 8
90*90	74.449	66.27	58.878	11.154 4	20.915
100*100	70.333	65.621	60.023	8.530 8	14.658 8
110*110	64.891	61.079	56.101	8.150 1	13.545 8
120*120	60.095	57.992	55.338	4.576 5	7.915 8
130*130	60.489	56.811	53.953	5.030 7	10.805 3
140*140	59.74	57.057	53.95	5.445 4	9.692
150*150	57.425	55.374	52.983	4.317 9	7.735 3
160*160	53.988	52.193	50.613	3.027 2	6.140 3
170*170	53.981	52.428	50.821	3.065 2	5.853 9
180*180	53.578	52.583	50.493	3.974 7	5.758 1
190*190	53.505	52.013	50.203	3.479 9	6.171 4
200*200	51.537	50.268	48.647	3.224 7	5.607 6

为了进一步验证 TSAANS 算法的性能和稳定性, 随机抽取 60 个轧制单元, 轧辊种类设定为 150 种, 运行 30 次, 记录每次的算法结果. 实验结果如图 4 所示.

通过以上实验可以得出如下结论:

1) 由表 1 可知, 在固定轧辊种类的情况下, TSAANS 算法在优化车削量上明显优于改进的人工算法, 平均相对优化比率高达 7%; 而且实验数据相同时, TSAANS 算法的优化效率也显著高于 BM-CA 算法, 在轧制单元数为 90 时, 其相对优化比率高达 14.4318%, 而且随着轧制单元数的增大, 优化空间越来越大.

2) BM-CA 和 TSAANS 算法之所以在优化车削量上明显优于改进的人工算法, 主要是由于 BM-CA 算法不仅考虑了车削轧辊匹配的顺序影响, 而且还考虑了交叉使用时车削量的优化; 同时, TSAANS 算法进一步在 BM-CA 算法基础上搜索轧制单元的替换邻域, 替换车削量较大的轧辊, 从而从整体上降低了轧制计划的车削成本.

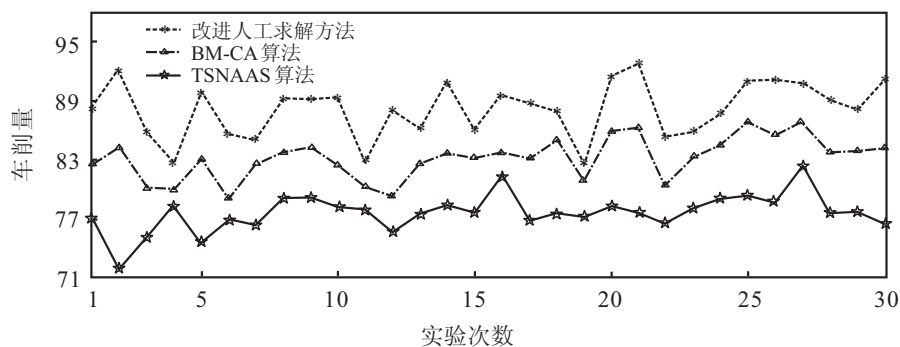


图 4 算法的性能和稳定性比较

3) 在固定轧制单元数、变动轧辊数量和种类的实验中, TSAANS 算法明显优于改进的人工算法和 BM-CA 算法 (见表 2). 相对于改进的人工算法, TSAANS 算法的平均优化比率高达 11.85%; 相对于第 1 阶段的 BM-CA 算法, 平均优化比率为 6.08%, 这在现实生产中是非常可观的.

4) 随着轧辊种类和数量的增加, 车削量稳定在 50~53 左右, BM-CA 算法和 TSAANS 算法的改进比率稳定在 3% 和 5% 左右, 即当轧辊种类数达到一定程度时, 无论是人工求解算法, 还是 BM-CA 算法和 TSAANS 算法, 轧制单元的车削量都会稳定在一个区间内, 增加轧辊种类将不能降低轧制计划的车削量. 这主要是由于随着轧辊的种类和数量的增大, 每个轧制单元内在匹配最佳车削轧辊时的选择机会增多, 更加容易选到最佳的轧辊, 导致后期的优化空间越来越小.

5) 由图 4 可知, BM-CA 算法和 TSAANS 算法具

有较高的性能和稳定性. 从 30 组随机实验中可以看出, TSAANS 算法的结果都优于改进人工算法和 BM-CA 算法. TSAANS 和 BM-CA 算法的轧辊车削量稳定在 71%~86% 之间, 具有较稳定的优化性能.

5 结 论

本文研究了一种考虑轧辊车削成本的轧制计划编制问题, 即在编制轧制计划时不仅考虑为每个轧制单元匹配最佳的车削轧辊, 而且考虑轧制单元之间的交叉使用. 通过深入分析问题特点及其复杂约束, 建立了以优化整个轧制计划的车削量为目标的 0-1 整数规划模型. 另外, 构造了一种基于替换邻域搜索的两阶段算法 (TSAANS): 第 1 阶段通过 BM-CA 算法依次为每个轧辊匹配车削轧辊, 同时为轧制计划实施交叉使用调整; 第 2 阶段通过 A-ANS 算法进一步搜索每个轧制单元的车削轧辊的替换邻域, 选取使整个轧制计划车削量最小的车削轧辊进行替换, 以降低整体车削

成本. 基于实际生产数据的实验结果表明, TSAANS 算法在优化车削量上对实际生产具有一定的指导意义.

参考文献(References)

- [1] 董广静, 李铁克, 王柏琳, 等. 考虑倒垛因素的轧制计划编制方法[J]. 控制与决策, 2015, 30(1): 149-155.
(Dong G J, Li T K, Wang B L, et al. Rolling planning problem with stacking factor[J]. Control and Decision, 2015, 30(1): 149-155.)
- [2] 王柏琳, 李铁克. 等待时间受限的流水车间调度问题的若干性质[J]. 系统工程理论与实践, 2012, 32(6): 1346-1352.
(Wang B L, Li T K. Some properties of the flowshop scheduling problem with limited waiting time constraints[J]. Systems Engineering—Theory & Practice, 2012, 32(6): 1346-1352.)
- [3] Lixin T, Lin H. Optimal and near-optimal algorithms to rolling batch scheduling for seamless steel tube production[J]. Int J of Production Economics, 2007, 105(2): 357-371.
- [4] 李耀华, 王伟, 徐乐江, 等. 热轧生产轧制计划模型与算法研究[J]. 控制与决策, 2005, 20(3): 275-279.
(Li Y H, Wang W, Xu L J, et al. Rolling plan model and algorithm in hot rolling plant[J]. Control and Decision, 2005, 20(3): 275-279.)
- [5] 张健欣, 童朝南. 用于编制热轧生产流程的新型 Meta-heuristics 算法[J]. 控制理论与应用, 2009, 26(7): 767-770.
(Zhang J X, Tong C N. A novel meta-heuristic algorithm for hot rolling scheduling[J]. Control Theory & Applications, 2009, 26(7): 767-770.)
- [6] 张涛, 王磊, 张玥杰. PSO 算法求解基于 PCVRP 的热轧批量计划问题[J]. 系统工程学报, 2010, 25(1): 55-61.
(Zhang T, Wang L, Zhang Y J. PSO algorithm for hot-milling batch planning problem based on PCVRP[J]. J of Systems Engineering, 2010, 25(1): 55-61.)
- [7] Lixin T, Jiyin L, Aiyang R, et al. A mathematical programming model for scheduling steelmaking-continuous casting production[J]. European J of Operational Research, 2000, 120(2): 423-435.
- [8] Peter C. A flexible decision support system for steel hot rolling mill scheduling[J]. Computers & Industrial Engineering, 2003, 45(2): 307-321.
- [9] Lixin T, Yang Y, Jiyin L. An efficient optimal solution to the coil sequencing problem in electro-galvanizing line[J]. Computers & Operations Research, 2010, 37(10): 1780-1796.
- [10] Jun Z, Wei W, Quanli L, et al. A two-stage scheduling method for hot rolling and its application[J]. Control Engineering Practice, 2009, 17(6): 629-641.
- [11] 贾树晋, 朱俊, 杜斌, 等. Pareto 最大最小蚂蚁算法及其在热轧批量计划优化中的应用[J]. 控制理论与应用, 2012, 29(2): 137-144.
(Jia S J, Zhu J, Du B, et al. Pareto max-min ant system algorithm and its application to hot rolling batch planning problem[J]. Control Theory & Applications, 2012, 29(2): 137-144.)
- [12] 屠乃威, 黄辉, 罗小川, 等. 热轧批量计划模型及其混合求解算法[J]. 系统工程学报, 2011, 26(1): 82-90.
(Tu N W, Huang H, Luo X C, et al. Model and hybrid algorithm for hot rolling lot planning[J]. J of System Engineering, 2011, 26(1): 82-90.)
- [13] 张文学, 李铁克. 基于粒子群和约束满足的轧钢一体化批量计划优化[J]. 计算机集成制造系统, 2010, 16(4): 840-846.
(Zhang W X, Li T K. Integrated batch planning optimization based on particle swarm optimization and constraint satisfaction for steel production[J]. Computer Integrated Manufacturing Systems, 2010, 16(4): 840-846.)

(责任编辑: 李君玲)