

加工时间可控的炼钢调度问题两阶段模型及优化算法

谭园园¹, 宋健海², 刘士新¹

(1. 东北大学 信息科学与工程学院 流程工业综合自动化教育部重点实验室, 辽宁 沈阳 110819;

2. 上海宝信软件股份有限公司, 上海 201900)

摘要: 炼钢-精炼-连铸是钢铁产品的关键生产工序, 其有效的调度对生产过程中减少热能消耗、提高生产效率具有重要意义. 根据生产过程中工序加工时间可控性和主要工艺约束提出了分散搜索(scatter search, SS)算法和数学规划相结合的两阶段求解算法. 第1阶段应用SS算法基于各阶段正常的加工时间, 确定炼钢-精炼生产阶段各设备的加工炉次集和各炉次的加工顺序. 第2阶段将SS求得的解转化为时间约束网络图, 建立了以炉次等待设备时间和设备等待炉次时间及最大完成时间最小为调度目标, 工序加工时间可控的混合整数规划模型, 应用CPLEX求解模型确定各炉次的加工时间和开始时间. 基于国内某钢铁企业炼钢-精炼-连铸生产过程的实绩生成了14个不同规模的测试案例, 对钢厂生产实绩效果与本文两阶段求解算法的优化效果进行了对比, 分析了不同等待时间权重对两阶段算法性能的影响, 并与采用遗传局域搜索(genetic local search, GLS)算法与数学规划相结合的求解算法的优化效果进行了比较. 实验结果表明本文给出的模型和两阶段求解算法对加工时间可控的炼钢-精炼-连铸调度问题的优化效果很好.

关键词: 炼钢-精炼-连铸调度; 加工时间可控; 分散搜索算法; 遗传局域搜索算法; 数学规划

中图分类号: C934 **文献标识码:** A

A hybrid two-phase algorithm and mathematical model for steelmaking and continuous casting with controllable processing time

TAN Yuan-yuan¹, SONG Jian-hai², LIU Shi-xin¹

(1. State Key Laboratory of Synthetical Automation for Process Industries, School of Information Science & Engineering, Northeastern University, Shenyang Liaoning 110819, China;

2. Shanghai Baosight Software Limited Company, Shanghai 201900, China)

Abstract: Steelmaking-refining-continuous casting is one of the key manufacturing processes in steel production, for which the optimal scheduling is an effective way for reducing the energy consumption and improving the production efficiency. By considering the required processing time and the technical constraints, we proposed a hybrid two-phase algorithm for the steel production, based on the scatter search (SS) method and the mathematical programming. In the first phase, the SS algorithm determines the order of the steelmaking-refining process and the technical sequence for each of them, based on the normal processing time. In the second phase, the solution obtained in the first phase is transformed into a temporal constraint network graph, and a mixed integer programming model with controllable processing time is built. The machine waiting time, the heat waiting time and the maximum completion time are minimized by using CPLEX. Totally, 14 different sets of randomly data collected from a Chinese iron and steel plant are used to test the model and the hybrid algorithm, and the results are compared with the practical results of the plant. The impact of the different weights for the waiting time on the effectiveness and efficiency of the hybrid algorithm is analyzed and compared with that of the combined genetic local search (GLS) algorithm and mathematical programming. Computational results show that the mathematical model and the two-stage algorithm are effective for solving the steelmaking-refining-continuous casting scheduling problem.

Key words: steelmaking-refining-continuous casting schedule; controllable processing time; scatter search algorithm; genetic local search algorithm; mathematical programming

1 引言(Introduction)

普通炼钢-精炼-连铸生产过程处于钢铁产品生产过程的上游, 是钢铁产品生产的关键环节, 其生产

过程的合理调度可以有效地提高企业的生产效率、降低生产成本和能源消耗. 在理论上, 通常将炼钢-精炼-连铸生产调度问题归结为具有NP-hard特性的

混合flow shop问题.

由于炼钢-精炼-连铸生产调度问题在理论研究和实际应用中的重要价值,一直吸引着国内外学者和业界人士的研究和关注^[1].文献[2]基于just-in-time思想给出了非线性规划模型,解决调度过程中的设备冲突问题,并转化为线性规划模型用标准线性规划程序对其进行求解.文献[3]给出了整数规划的调度模型,设计了拉格朗日松弛、动态规划和启发式算法相结合的求解算法.文献[4]以实际钢铁企业为研究背景,给出了混合整数规划模型,应用标准软件包对其进行求解.文献[5]设计了蚁群优化和非线性优化相结合的混合算法,第1阶段确定工件的加工设备及工件的加工顺序,第2阶段应用非线性优化方法确定工件的加工开始时间.文献[6]建立了0-1混合整数规划模型,对模型进行Benders分解,提出了第1阶段应用遗传算法提高种群质量和第2阶段应用线性规划指导遗传算法迭代的两阶段求解算法.文献[7]建立了带运输时间的两机器调度模型和炼钢-精炼两阶段多并行机的调度模型,其中两阶段调度模型以最小化最大完成时间、设备空闲时间和炉次等待设备时间为调度目标,设计了禁忌搜索算法对模型进行求解.目前,对加工时间可控的调度问题的研究不多,文献[8]给出了工序加工时间可控的排列流水车间(permutation flow shop)调度问题的模型,并设计了结合插入算子的局域搜索算法.文献[9]采用三角模糊数表示炼钢连铸生产中工序的不确定加工时间,建立了模糊规划调度模型并设计了遗传算法对其进行求解.

分析以上研究成果作者发现,现有研究一般均假设钢水在各设备上的加工时间为固定值,并把炼钢-精炼-连铸生产调度问题转化为以makespan最小为优化目标的混合flow shop问题进行求解.这种假设与炼钢-精炼-连铸实际生产过程存在以下差异:1)在实际生产过程中,钢水在连铸设备上的连续浇铸是生产调度人员优先保证的作业目标,为了实现连续浇铸目标,调度人员可以根据生产实绩在一定范围内动态调整钢水在各设备上的加工时间,即钢水在各设备的加工时间可控;2)炼钢-精炼-连铸加工过程中钢水在各阶段的工艺温度必须在一定的范围内,因此,钢水在设备间的传搁时间存在上下限要求,并且要求连铸工序连续加工;3)为了提高生产效率,降低能源消耗,生产调度的目标不仅要考虑makespan最小,还要最小化钢水在设备间的等待时间以及设备的空闲时间.上述研究成果中,文献[4,6]以炉次的最大完工时间、浇次的开浇提前/托期惩罚最小为调度目标,未考虑炉次等待设备时间的调度目标;文献[2-3,7]同时考虑了炉次等待时间和设备等待时间最小的调度目标,但均假定工序的

加工时间是固定的,且文献[7]仅考虑了炼钢和精炼两阶段的调度计划模型.文献[8]考虑了permutation flow shop调度问题中工序加工时间的可控性,没有考虑炼钢-精炼-连铸加工过程中钢水在设备间传搁时间的上下限要求及连铸工序连续加工的特性.文献[9]考虑了炼钢连铸调度问题中加工时间不确定的特点,但并未将炉次等待设备时间和设备等待炉次时间作为优化目标.

本文针对问题实际特点及现有研究方法的不足,以最小化makespan、最小化炉次等待时间和设备等待时间之和为优化目标,提出了炼钢-精炼-连铸生产调度计划的两阶段优化方法.第1阶段以各炉次在各阶段的正常加工时间为参数,采用分散搜索(scatter search, SS)算法确定各炉次在各阶段的加工设备和各设备上的炉次加工顺序.第2阶段将第1阶段求得的可行解转化为时间约束网络图,综合考虑makespan、炉次等待时间和设备等待时间最小的目标,建立工艺路径已知、加工顺序已知、加工时间可控的炼钢-精炼-连铸调度问题的混合整数规划模型,应用优化软件(CPLEX)求解确定各工序的开始时间和加工时间.

本文其余部分组织结构如下:第2节给出了加工时间可控的炼钢-精炼-连铸调度问题的描述;第3节给出了SS算法和数学规划算法相结合的两阶段优化算法,详细介绍了SS算法的基本算子的实现过程,并基于时间约束网络图建立了炼钢-精炼-连铸调度问题的混合整数规划模型;第4节针对随机生成测试案例和国内某炼钢厂生产实绩数据的实验结果进行对比分析;最后给出了本文的结论.

2 问题描述(Problem description)

炼钢-精炼-连铸调度可具体描述成如下混合flow shop调度问题: n 个工件(炉次) $j = \{1, \dots, n\}$,分别要顺序地通过炼钢、精炼、连铸生产阶段加工,工件 j 包含工序 $\{O_{1j}, O_{2j}, \dots, O_{l_{\max}j}\}$,其中 $l_{\max}$ 为最大工序阶段数.在第 l 个生产阶段有 m_l 台相同的设备可供选择,同一工件在同一时间只能在一个设备上加工,同一设备在同一时间只能加工一个工件.工件 j 在炼钢或精炼生产阶段加工完成后,由运输设备运输到下一生产阶段的目的设备,如果目的设备被占用,则工件 j 需要等待直到目的设备空闲.工件在运输及等待设备的同时伴随温度下降和热能损失.炼钢-精炼-连铸生产过程要求钢水温度保持在一定的范围,因此,工件在设备间运输及等待时间必须控制在一定的范围内,即大于等于最小运输时间、小于等于最大允许温降时间.如果当前设备上的工件加工完成,下一个待加工的工件未到达,会出现加工设备等待加工工件的现象,等待时间过长会明显降低设备的利用率.在连铸阶段为了提高材料收得

率、降低生产成本,要求属于同一浇次的工件按给定顺序在同一连铸设备上连续加工,而且后继工件必须在当前工件完成加工前到达浇铸工位.同一连铸机加工的相邻两浇次间必须有一定的设置时间.由于存在各种不同程度的扰动因素,为了保证连铸加工阶段的钢水连续浇铸,可以在一定的范围内调整工件在各生产阶段的加工时间.调度任务是确定炼钢、精炼阶段各设备的加工工件集合、各设备上工件的加工顺序和各工序的加工时间,使得makespan最小、工件等待时间和设备等待时间之和最小,从而降低生产能耗、提高设备利用率.

3 两阶段优化方法(Two-stage optimal algorithm)

3.1 阶段1: 确定工件加工设备及加工顺序(First stage: determine processing machine and job sequence)

炼钢-精炼-连铸调度问题是集指派和排序特点的一类复杂的组合优化问题,连铸阶段的连浇要求和传搁时间的上下限约束增加了问题的求解难度.因此,本文设计了SS算法. SS算法由Glover于1977年提出的一种基于种群进化的启发式算法^[10],一些学者已成功应用于求解各类组合优化问题和非线性规划问题^[11-14]. SS算法的基本过程描述如下: 1) 生成初始种群,用于初始化参考集; 2) 选择初始种群中的最优解与多样性解构成参考解集; 3) 选择参考集中的两个或多个解构成若干个子集; 4) 对子集中的解应用组合算子,生成新解; 5) 应用局域搜索算子提高新解的质量并对参考集进行更新. SS算法的基本组成部分包括: 初始解生成方法、参考集更新方法、子集生成方法、组合算子、局域搜索算子.

3.1.1 解的编码(Encoding of a solution)

根据SS算法属于元启发式算法的特性和第2节对炼钢-精炼-连铸调度问题的描述,本文采用整数编码. 记 $\pi = (\pi_1, \dots, \pi_{l_{\max}})$ 表示调度问题的可行解,其中 $\pi_l = \{\pi_{l1}, \dots, \pi_{lm_l}\}$ 表示阶段 l 所有设备的加工序列, $\pi_{li} = \{\pi_{li}(1), \pi_{li}(2), \dots, \pi_{li}(n_{li})\}$ 为阶段 l 加工设备 i 的加工工序集合,其中 $\pi_{li}(k)$ 表示阶段 l 机器 i 第 k 个位置加工的炉次.

3.1.2 初始解的生成(Initial solutions generation method)

在炼钢-精炼-连铸生产组织过程中,各炉次在连铸阶段的加工设备和加工顺序是事先给定的.因此,本阶段SS算法只需要根据在连铸阶段确定的炉次加工设备和加工顺序来优化各炉次在炼钢、精炼生产阶段的加工设备和加工顺序.本文采用分阶段倒序方式生成初始解.首先,根据设备平均利用率相等的原则,随机生成精炼阶段每台设备的加工工序集.其

次,根据连铸阶段各设备的加工工序集和加工顺序倒推计算各炉次在连铸阶段的加工开始时间,将精炼设备的工序集按各炉次的开浇时间升序排列,得到精炼阶段的加工顺序.炼钢阶段各设备的加工工序集和加工顺序可同样得出.本文给出的初始解生成方法保证了各阶段设备利用率均衡,同时使加工阶段间的加工序列衔接紧密,减少了因等待设备引起的热能损失.

3.1.3 组合算子(Combination operator)

组合算子是SS算法的基本组成部分,合理设计组合算子即可提高解的优质性又可保证解的多样性.本文根据炼钢-精炼-连铸调度问题特点及编码方式采用标准两点顺序交叉的方法.下面以精炼阶段各设备加工工件集合的产生过程为例描述组合算子的基本流程.

1) 重组.

以设备为单元,将精炼阶段加工工件集合重组成长度为 $(n + m_l)$ 的一维数组,相邻设备的加工工件集合间插入设备编号($n + i$ 代表该加工阶段的第 i 台设备).重组过程如图1所示.

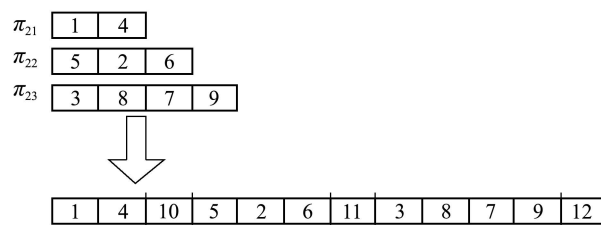


图1 重组精炼设备加工的工件

Fig. 1 Recombining jobs processed on refining devices

2) 两点顺序交叉.

随机产生父代中的两个交叉点,交换两点间的工作.从第2个交叉点后第1个基因起列出原顺序,到达个体尾时返回头继续,去掉已有基因,将获得的无重复顺序填入新个体.交叉过程如图2所示.

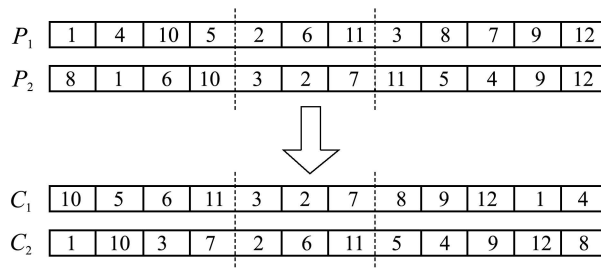


图2 两点顺序交叉

Fig. 2 Two-points order crossover

3) 分解.

顺序交叉后产生的个体,根据设备编号对其进行分解.连续找出子代个体中的两个设备编号 X, Y ,

设备编号间的工件为设备 $Y-n$ 的加工工件集. 直到找到最后一个设备编号. 最后一个设备编号后的工件序列和第1个设备编号前的工件序列为第1个设备编号代表的设备的加工工件集. 下面以子代 C_2 解码过程进行说明, 如图3所示.

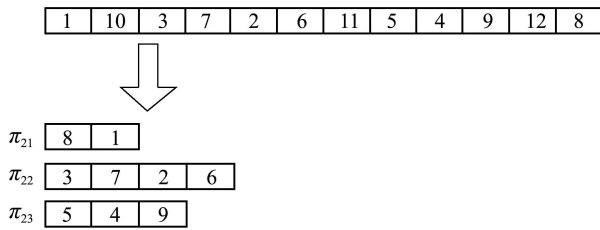


图3 分解过程

Fig. 3 Decomposition process

4) 排序.

对产生的精炼设备的加工工件集合按工件在连铸阶段加工开始时间进行升序排序. 如图4所示.

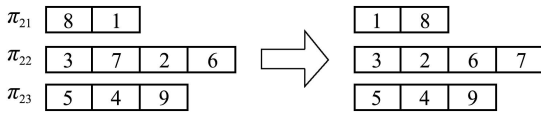


图4 排序过程

Fig. 4 Sorting process

炼钢阶段解的产生方法类似. 顺序交叉较好地

保留了相邻关系、先后关系, 满足炼钢-精炼-连铸调度问题的需要.

3.1.4 局域搜索算子(Local search operator)

局域搜索算子对初始种群中的解和组合算子产生的解进行操作, 通过搜索邻域内最优解提高解的质量. 为了更好的描述局域搜索算子, 引入文献[15]中的block概念. 关键路径包括 $l_{\max}$ 个子路径, 第 l 个子路径包括从最小位置 e_l 到最大位置 f_l 间的工件序列. 即点的序列表示为

$$(1, \pi_{1h_1}(e_1)), \dots, (1, \pi_{1h_1}(f_1)), \dots, (l_{\max}, \pi_{l_{\max}h_{l_{\max}}}(e_{l_{\max}})), \dots, (l_{\max}, \pi_{l_{\max}h_{l_{\max}}}(f_{l_{\max}})),$$

其中: $\pi_{lh_l}(f_l) = \pi_{l+1, h_{l+1}}(e_{l+1}) (l=1, \dots, l_{\max}-1)$ 表示相同工件在相邻阶段的加工. 关键路径上的第 l 个子路径

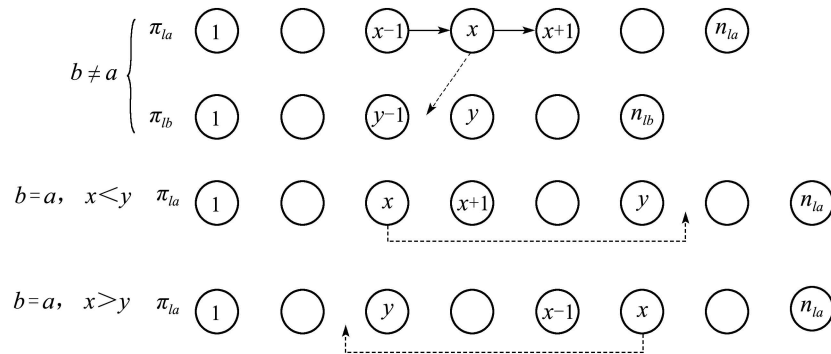
$$(l, \pi_{lh_l}(e_l)), (l, \pi_{lh_l}(e_l+1)), \dots, (l, \pi_{lh_l}(f_l))$$

的工件序列

$$B_l = (\pi_{lh_l}(e_l), \pi_{lh_l}(e_l+1), \dots, \pi_{lh_l}(f_l))$$

构成了阶段 l 上的一个block. b_l 是block B_l 上的工件个数, $b_l = |B_l| = f_l - e_l + 1, l=1, \dots, l_{\max}$.

在本文的SS算法中, 邻域解是通过插入类型的移动产生的. l 为所选的生产阶段, a, b 表示该阶段的两个设备, x, y 分别是向量 π_{la}, π_{lb} 的位置变量. 向量 $v = (l, a, x, b, y)$ 表示移动向量, 具体表现为工件 $\pi_{la}(x)$ 从集合 π_{la} 的第 x 个位置移走, 然后插入到集合 π_{lb} 中的第 y 个位置. 具体移动形式如图5所示.

图5 对移动 $v = (l, a, x, b, y)$ 的说明Fig. 5 Illustration of move $v = (l, a, x, b, y)$

通过移动 v 的作用, 产生新的解 π_v . π 的邻域由一系列根据移动 v 产生的 π_v 构成, 即 $\{\pi_v : v \in V(\pi)\}$, 相应的移动集合

$$V(\pi) = \bigcup_{l=1}^{l_{\max}} \bigcup_{a=1}^{m_l} \bigcup_{x=1}^{n_{la}} \bigcup_{b=1}^{m_l} V_{laxb}(\pi),$$

其中:

$$V_{laxb}(\pi) =$$

$$\begin{cases} (l, a, x, b, y) : 1 \leq y \leq n_{lb}, y \notin \{x-1, x\}, & \text{如果 } b = a, \\ (l, a, x, b, y) : 1 \leq y \leq n_{lb} + 1, & \text{如果 } b \neq a, \end{cases}$$

$a, b = 1, \dots, m_l, x = 1, \dots, n_{la}, l = 1, \dots, l_{\max}$, 集合 $V_{laxb}(\pi)$ 包含的移动为工件 $\pi_{la}(x)$ 从 π_{la} 中的第 x 个位置移走, 然后插入到序列 π_{lb} 所有可能的位置.

为了提高邻域的搜索速度, 应尽可能的剔除

邻域集合里一些对目标值没有改进的解, 根据文献[15]中给出的剔除无效移动的规则, 缩减后的移动集合 $V(\pi)$ 表示为

$$W(\pi) = \bigcup_{l=1}^{l_{\max}} \bigcup_{x=e_l}^{f_l} \bigcup_{b \in M_l} W_{lh_lxb}(\pi),$$

对于阶段 l , 缩减后的移动集合 $W_{lh_lxb}(\pi) \subseteq V_{lh_lxb}(\pi)$, $b \in M_l$, $x = e_l, \dots, f_l$ (这里假设 $a = h_l$), 则

$$W_{laxb}(\pi) = \begin{cases} V_{laxb}(\pi), & \text{如果 } b \neq a, e_l < f_l, \\ V_{laxa}(\pi) \setminus \bigcup_{y=e_l+1}^{f_l-1} \{(1, a, x, a, y)\}, & \text{如果 } b = a, e_l < x < f_l, \\ V_{laxa}(\pi) \setminus \bigcup_{y=1}^{e_l} \{(1, a, x, a, y)\}, & \text{如果 } b = a, x = e_l \neq f_l, l \neq 1, \\ V_{laxa}(\pi) \setminus \bigcup_{y=f_l}^{n_{la}} \{(1, a, x, a, y)\}, & \text{如果 } b = a, x = f_l \neq e_l, l \neq m, \\ \varphi, & \text{如果 } e_l = f_l. \end{cases}$$

梯度权重表示炉次在连铸和精炼设备上的加工顺序的权重值, 连铸和精炼阶段的梯度权重分别以浇次和设备为单位进行梯度计算. 如: $\pi_{31} = \{1, 3, 5, 6\}$, $\pi_{32} = \{2, 4, 7, 8\}$, 浇次序列为 $\{1, 3\}$, $\{5, 6\}$, $\{2, 4\}$, $\{7, 8\}$, 则8个炉次的梯度权重为 $w_3 = \{1, 1, 2, 2, 9, 10, 9, 10\}$, 同一设备加工的, 属于不同浇次的炉次权重值是按工件数 n 梯度增长的.

定理 1 如果最大等待时间

$$0 < W < \min\{p_{lj} + p_{l+1,j}\},$$

则在以 makespan 最小为优化目标的最优解中, 对于同一连铸机加工的炉次, 在精炼阶段同一设备上的加工顺序按浇铸阶段的梯度权重升序排列.

证 反证法. 对于同一连铸机加工的炉次, 在精炼阶段同一设备上的加工顺序按浇铸阶段的梯度权重降序排列.

1) 炉次属于同一梯度.

同一梯度即表示两个炉次属于同一浇次. 假设存在同一连铸机上加工的连续两炉次 $\pi_{31}(j) = q_1$, $\pi_{31}(j+1) = q_2$, 在精炼阶段同一加工设备上, 两炉次的加工顺序按梯度权重降序排列为 $\pi_{21}(k) = q_2$, $\pi_{21}(k+1) = q_1$. 根据同一浇次连浇的约束条件, q_2 的浇铸开始时间 $s_{3,q_2} = s_{3,q_1} + p_{3,q_1}$, q_1 在精炼阶段的加工开始时间 $s_{2,q_1} \geq s_{2,q_2} + p_{2,q_2}$, 那么 q_2 的等待时间 $w_{3,q_2} \geq p_{2,q_1} + p_{3,q_1} \geq \min\{p_{lj} + p_{l+1,j}\} > W$, 与等待时间不能超过给定上限值的约束矛盾.

2) 炉次属于不同梯度.

不同梯度即表示两个炉次属于不同浇次. 假设存在同一连铸机上加工的连续两炉次 $\pi_{31}(j) = q_1$, $\pi_{31}(j+1) = q_2$, 且属于不同浇次, 在精炼阶段同一加工设备上, 两炉次的加工顺序按梯度权重降序排列为 $\pi_{21}(k) = q_2$, $\pi_{21}(k+1) = q_1$. 基于不同浇次间存在设置时间的约束, q_2 的浇铸开始时间 $s_{3,q_2} = s_{3,q_1} + p_{3,q_1} + S$. q_1 在精炼阶段的加工开始时间 $s_{2,q_1} \geq s_{2,q_2} + p_{2,q_2}$, 那么 q_2 的等待时间 $w_{3,q_2} \geq p_{2,q_2} + p_{3,q_2} + S > \min\{p_{lj} + p_{l+1,j}\} > W$, 与等待时间不能超过给定上限值的约束矛盾, 故上面假设不成立. 证毕.

推论 1 如果最大等待时间

$$0 < W < \min\{p_{lj} + p_{l+1,j}\},$$

则在以 makespan 最小为优化目标的最优解中, 对于同一精炼设备加工的炉次, 在炼钢阶段同一设备上的加工顺序按精炼阶段的梯度权重升序排列.

根据定理 1 和推论 1, 移动产生的解必须保证按炉次的梯度权重升序排列, 在缩减移动集中剔除移动后炉次按梯度权重降序排列的移动. 梯度权重既保证了当前阶段的炉次顺序与后继阶段同一设备炉次加工顺序一致, 减少了设备等待炉次时间和热能损失, 又可实现同一阶段不同设备间炉次排列顺序的任意性, 保证了解的多样性.

3.2 阶段2: 确定工件的开始加工时间及加工持续时间 (Second stage: determine starting time and processing time of jobs)

3.2.1 可行解转化为时间约束网络图 (Temporal constraint network graph)

第1阶段求得的可行解可以转换为一张时间约束网络图. 以4个炉次3个生产阶段, 每个生产阶段设置2个生产设备为例, 应用SS算法求解后, 产生的可行解 $\{\{1, 2\}, \{3, 4\}\}, \{\{3, 2\}, \{1, 4\}\}, \{\{1, 2\}, \{3, 4\}\}$, 转化为时间约束网络后如图6所示.

图6中节点代表各炉次的不同工序, 其中节点0和节点*为虚工序, 分别代表整个调度计划的开始和结束. 节点上面的数据代表对应工序的[最小加工时间, 最大加工时间]范围, 节点间的弧线代表各炉次不同工序间的时间约束关系, 分为以下3种:

1) 同一设备不同工序间的时间约束关系, 通过虚线弧连接表示, 箭头方向代表加工顺序, 对应 $[0, +\infty)$ 时间约束, 表示前一工序加工完成后下一工序才能开始. 如图6所示, 工序 O_{11} , O_{12} 被分配在第1生产阶段的同一台设备上, 加工顺序为 $O_{11} \rightarrow$

O_{12} , 则它们间存在虚线弧 $[0, +\infty]$ 连接.

2) 同一炉次的不同工序在不同生产阶段上加工的时间约束关系, 通过实线弧连接表示, 对应 $[t_1, t_2]$ 时间约束, 表示工件在上一设备加工完成后到下一设备开始的时间大于等于设备间的运输时间 t_1 , 小于等于最大等待时间 t_2 , t_2 由加工温度等工艺约束决定. 如图6所示, 工件1的工序 O_{11} 完成后要转移到第2个阶段的某台设备上加工工序 O_{21} , 则它们间存在连接弧 $[3, 20]$ 连接.

3) 连铸阶段同一设备上不同工序间的时间约束关系, 通过实线弧连接表示, 对应 $[t_1, t_2]$ 时间约束, 表示上一工件加工完成后到下一工件到达的时间大于 t_1 , 小于等于 t_2 , 这里 t_1, t_2 由连铸工艺约束决定. 对于相同浇次的相邻两炉次, 后继炉次必须在前继炉次加工完成前若干分钟到达. 如图6中的 O_{31} 到 O_{32} 间的时间约束 $[-5, 5]$, 表示炉次2必须在炉次1加工完成前5分钟到达该工位. 不同浇次的相邻两炉次间有一定的设置时间间隔.

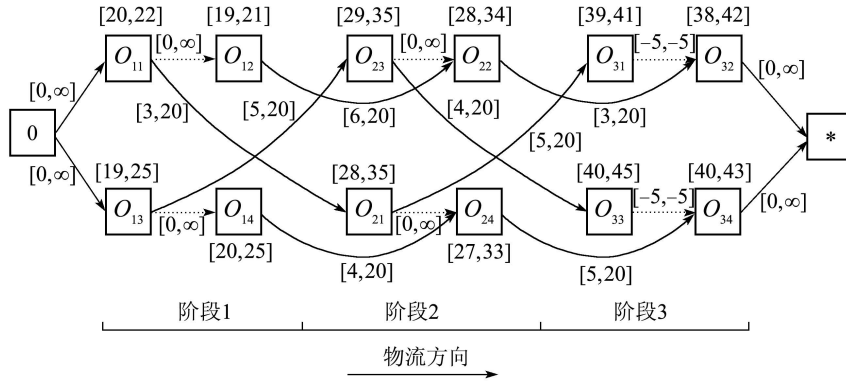


图6 可行解对应的时间约束网络

Fig. 6 Temporal constraint network graph of the feasible solution

3.2.2 符号说明(Explanation of symbols)

规定以下符号:

l : 阶段号, $l = 1, \dots, l_{\max}$;

j : 工件编号, $j = 1, \dots, n$;

O_{lj} : 工件 j 在阶段 l 对应的工序;

C : 同一炉次的工序间先后关系的弧集(实线弧集);

D : 不同炉次但被分配到同一设备的工序间先后关系的弧集(虚线弧集);

$P_{lj}^{\min}$: 工序 O_{lj} 的加工时间最小值;

$P_{lj}^{\max}$: 工序 O_{lj} 的加工时间最大值;

$d_{lj,l+1j}^{\min}$: 工序 O_{lj} 完成至工序 $O_{l+1,j}$ 开始的最小时间间隔即运输时间;

$d_{lj,l+1j}^{\max}$: 工序 O_{lj} 完成至工序 $O_{l+1,j}$ 开始的最大时间间隔即最大等待时间;

Ca_j : 工件 j 的浇次号;

S : 相邻浇次间的设置时间;

Pr : 同一浇次炉次间的准备时间;

p_{lj} : 工序 O_{lj} 的加工时间, 为决策变量;

t_{lj} : 工序 O_{lj} 的开始加工时间, 为决策变量.

3.2.3 数学模型(Mathematical model)

$$\min t_{l_{\max}+1,*}, \quad (1)$$

$$\min \left\{ \sum_{((l,j),(l+1,j)) \in C} (t_{l+1,j} - t_{lj} - p_{lj} - d_{lj,l+1j}^{\min}) + \sum_{((l,i),(l,j)) \in D} (t_{lj} - t_{li} - p_{li}) \right\}, \quad (2)$$

$$\text{s.t.} \begin{cases} d_{lj,l+1j}^{\min} \leq t_{l+1,j} - t_{lj} - p_{lj} \leq d_{lj,l+1j}^{\max}, \\ ((l,j),(l+1,j)) \in C, \end{cases} \quad (3)$$

$$\begin{cases} t_{lj} - t_{li} \geq p_{lj}, ((l,i),(l,j)) \in D, \\ t_{lj} = t_{li} + p_{li}, \end{cases} \quad (4)$$

$$\begin{cases} ((l,i),(l,j)) \in D, l = l_{\max}, Ca_i = Ca_j, \\ t_{lj} + p_{lj} \geq t_{l-1,i} + p_{l-1,i} + d_{l-1,i,lj}^{\min} + Pr, \end{cases} \quad (5)$$

$$\begin{cases} ((l,i),(l,j)) \in D, l = l_{\max}, Ca_i = Ca_j, \\ t_{lj} \geq t_{li} + p_{li} + S, \end{cases} \quad (6)$$

$$((l,i),(l,j)) \in D, l = l_{\max}, Ca_i \neq Ca_j, \quad (7)$$

$$p_{lj}^{\min} \leq p_{lj} \leq p_{lj}^{\max}, j = 1, 2, \dots, n, \quad (8)$$

其中: 目标函数(1)表示所有工件的最大完成时间最小; 目标函数(2)表示各炉次等待设备时间与设备等待炉次时间之和最小; 约束式(3)表示同一炉次的相邻工序间时间间隔大于运输时间小于最大等待时间; 约束式(4)表示分配在同一设备上的两个炉次不能同时加工; 约束式(5)表示在连铸阶段同一浇次的相邻两炉次必须满足连浇约束; 约束式(6)表示在连铸阶段工件必须在加工开始前

一给定时间内到达; 约束式(7)表示连铸阶段属于不同浇次的相邻两炉次间有一定的设置时间; 约束式(8)表示决策变量的取值范围。

基于CPLEX不能求解多目标问题的特点, 本文采用加权的方式将上述给出的多目标数学模型转化为单目标。加权的方法能充分体现不同钢铁企业对工件的完工时间、工件等待时间和设备等待时间的重视程度不同, 即目标的权重大小不一。

4 实验结果(Experimental results)

4.1 对照算法(Comparative algorithm)

实验中, 为了比较SS算法和数学规划方法相结合的两阶段算法对本文问题的求解性能, 本文作者设计了遗传局域搜索(genetic local search, GLS)算法和数学规划算法相结合的求解算法。GLS算法采用与SS相同的编码的方式。交叉算子采用单点交叉。当满足交叉概率时, 对父代个体的每一生产阶段 $l(l = 1, \dots, l_{\max})$ 执行如下的交叉过程: 首先, 将两父代个体当前生产阶段 l 各设备加工的工件集合重组成长度为 $(n + m_l)$ 的一维数组; 其次, 随机生成交叉点, 对两父代个体重组后的排列执行单点交叉过程; 再次, 对交叉后产生的新的工件和设备的排列, 根据设备编号进行分解得到各生产设备加工的工件集合; 最后, 对交叉产生的子代个体按后继阶段的开始时间进行升序排列, 即确定各设备加工工件的顺序。当满足变异概率时, 对父代个体每一生产阶段 $l(l = 1, \dots, l_{\max})$ 执行如下的变异过程: 首先, 对阶段 l 设备加工的工件集合进行重组; 其次, 随机生成两个变异结点位置, 对换变异位置的值。再次, 执行同交叉算子中相同的分解和排序过程。将交叉算子和变异算子产生的新个体插入到新种群, 对其中的个体执行3.1.4节的局域搜索算子。选择算子采用最优和轮盘赌方法, 在产生的新种群与父代种群中优先选择最好的个体复制到下一代, 其余个体采用轮盘赌方法产生。将GLS生成的最优解应用3.2节介绍的数学规划方法进行求解确定各工件的开始时间及加工时间。

4.2 实验设计及问题参数设置(Experimental design and parameter settings)

炼钢-精炼-连铸调度问题没有标准的测试数据, 本文分别采用国内某炼钢厂生产实绩数据和根据生产实绩数据随机生成的测试案例对模型及两阶段算法进行测试。该炼钢厂炼钢、精炼、连铸阶段各有2台加工设备, 炼钢与精炼间的

传搁时间取自区间 $[7, 15]$, 精炼与连铸的传搁时间为 $[5, 7]$ 内一定值。随机生成案例参数设置: 设各工件的最大等待时间均相等即 $d_{ij,l+1j}^{\max} = 20 \text{ min}$, $j = 1, \dots, n$, $l = 1, \dots, l_{\max} - 1$, 设置时间 $S = 5 \text{ min}$, 准备时间 $Pr = 3 \text{ min}$ 。工件在各生产阶段的最大、最小加工时间在表1给出的区间内随机产生, 不同工件的加工时间区间[最小加工时间, 最大加工时间]不同, 可全面的测试本文给出的两阶段算法和数学模型优化效果和时间效率。

表1 各阶段工序最大、最小加工时间区间

Table 1 Minimum and maximum processing time interval

l	1		2		3	
	$p_1^{\min}$	$p_1^{\max}$	$p_2^{\min}$	$p_2^{\max}$	$p_3^{\min}$	$p_3^{\max}$
	(35, 40)	(50, 55)	(15, 20)	(55, 60)	(35, 40)	(50, 55)

根据每一浇次包含的炉次数以10为界, 生成不同规模的两组测试案例: 工件数 $n_1 = \{32, 48, 54\}$, $n_2 = \{45, 66\}$ 。设备组合 $m_1 * m_2 * m_3 = \{2 * 2 * 2, 3 * 3 * 2, 3 * 3 * 3, 3 * 4 * 3, 3 * 5 * 3, 3 * 6 * 3\}$, 组合后的测试案例分别为 $(\{2 * 2 * 2, 3 * 3 * 2\})$, $(\{32, 45, 54\})$, $(\{3 * 3 * 3, 3 * 4 * 3, 3 * 5 * 3, 3 * 6 * 3\})$, $(\{48, 66\})$, 除工件数 $n_1 = 54$, 每台连铸机加工3个浇次外, 其余案例中每台连铸机包含2个浇次, 则一共生成14个测试案例。

SS算法参数设置如下: 算法的终止条件为最大迭代代数 $\max \text{Iter} = 100$ 或最优解的目标值连续30代没有变化, 初始种群 P 的大小为100, 参考集规模 $b = 10$ 。在实验中, GLS算法参数设置如下: 算法的终止条件为最大迭代代数为100或最优解的目标值连续30代没有变化, 种群大小为20, 单点交叉概率为0.85, 变异概率为0.1。

4.3 两阶段算法对等待时间的优化效果 (Effectiveness of two-stage algorithm for waiting time)

本节主要对第1阶段应用SS算法和第2阶段应用CPLEX求解的等待时间进行对比分析。SS算法中工件在当前阶段加工完成后马上运输到下一阶段进行加工, 即工件开始时间仅考虑当前工件的等待时间最小, 未从全局角度考虑多个工件间的相互协调。CPLEX求解阶段基于SS算法给出的工件加工设备和工件加工顺序, 利用加工时间可控性的特点, 综合考虑工件间的相互协调以进一步降低工件的等待时间和设备等待时间。记第1阶段

SS求得的最优解对应的等待时间为 $W1$, 第2阶段采用CPLEX计算的等待时间为 $W2$, 表中比值 $W2/W1$.

CPLEX求得的等待时间与SS算法求得的等待时间的比值小说明两点: 1) 炉次等待设备时间少, 即因

炉次等待引起的热能损失少; 2) 设备等待炉次时间少, 即各加工阶段的设备利用率高. 实验中应用两阶段算法对每个测试案例求解10次, 统计每个实例每次的等待时间比值, 以及10次比值的平均值. 统计结果如表2所示.

表2 两阶段算法最优解对应的等待时间

Table 2 Waiting time of optimal solutions obtained by the two-stage algorithm

Num	$n = 32(2)$		$n = 54(3)$		$n = 48(2)$				$n = 45(2)$		$n = 66(2)$			
	2*2*2	3*3*2	2*2*2	3*3*2	3*3*3	3*4*4	3*5*3	3*6*3	2*2*2	3*3*2	3*3*3	3*4*4	3*5*3	3*6*3
1	0.0089	0.0134	0.1007	0.1583	0.2069	0.2104	0.1816	0.1645	0.1049	0.0855	0.1902	0.2454	0.2467	0.2455
2	0.0692	0.0882	0.1504	0.2739	0.2098	0.2232	0.1881	0.1921	0.1657	0.1370	0.2537	0.2581	0.2513	0.2522
3	0.1310	0.1361	0.1790	0.2868	0.2128	0.2604	0.2169	0.2083	0.1701	0.1461	0.2546	0.2592	0.2556	0.2676
4	0.1476	0.1496	0.2151	0.3061	0.2268	0.2753	0.2387	0.2300	0.1789	0.2847	0.2757	0.2794	0.2733	0.2795
5	0.1556	0.1645	0.2232	0.3174	0.2292	0.2885	0.2455	0.2502	0.1821	0.2938	0.2885	0.2820	0.2770	0.2799
6	0.1599	0.1709	0.2290	0.3214	0.2345	0.2892	0.2706	0.2698	0.1896	0.2967	0.2917	0.2850	0.3152	0.3106
7	0.1717	0.1807	0.2302	0.3267	0.2570	0.2893	0.2755	0.2961	0.1979	0.3022	0.3020	0.3026	0.3162	0.3238
8	0.1779	0.2436	0.2329	0.3316	0.2576	0.2927	0.2762	0.3261	0.2011	0.3041	0.3021	0.3051	0.3204	0.3240
9	0.1885	0.2641	0.2337	0.3342	0.2669	0.2953	0.2819	0.3323	0.2139	0.3052	0.3066	0.3108	0.3232	0.3682
10	0.2124	0.2895	0.2441	0.3387	0.2749	0.2962	0.2820	0.3387	0.2324	0.3347	0.3150	0.3397	0.3245	0.3694
Avg	0.1423	0.1700	0.2038	0.2995	0.2376	0.2721	0.2457	0.2608	0.1837	0.2490	0.2781	0.2867	0.2903	0.3021

根据表2可以得出以下结论: 1) 相同设备组合下, 随着工件数的增多, SS阶段和CPLEX阶段的最优解对应的等待时间比值增大. 如设备组合为 $2*2*2$ 时, 工件数 $n = 32$ 的等待时间比值的平均值明显比 $n = 45$ 的平均比值小. 2) 等待时间比值与每台连铸机上加工的浇次数量有关, 浇次数越多等待时间比值越大. 设备组合为 $2*2*2$ 时, 对2个浇次 $n = 32$ 的测试案例和3个浇次 $n = 54$ 的实例进行比较, 后者等待时间比值比前者明显大很多. 3) 连铸阶段设备数相同时, 等待时间比值与炼钢和精炼设备的数量成反比关系. 如 $n = 32$ 时, 设备组合 $3*3*2$ 的等待时间比值比设备组合 $2*2*2$ 的等待时间比值大. 4) 对于炼钢和连铸设备数相等的情况下, 精炼设备越多等待时间比值越大. 当 $n = 48$ 时, 等待时间比值依 $m_2 = 3, 4, 5, 6$ 呈递增

趋势. 因此, 等待时间与工件数, 浇次数和各阶段的加工设备数有关. 从对等待时间的优化效果的角度考虑, CPLEX的求解结果更能达到减少炉次等待时间和设备等待时间的目的.

4.4 两阶段算法对钢厂计划优化效果分析
(Optimal effectiveness on steel plan of two-stage algorithm)

本节对应用两阶段优化算法求解的某钢厂生产计划的结果与生产实绩数据进行对比分析. 实验结果如表3所示, 其中cast size给出了生产计划中每台连铸机加工的浇次数和每个浇次包含的炉次数, 如 $\{\{15, 11\}, \{10, 13\}\}$ 表示每台连铸机各加工2个浇次, 第1台连铸机的第1个浇次包含15个炉次. JWT, MWT分别表示炉次等待设备时间和设备等待炉次时间.

表3 两阶段算法的优化效果与实绩结果

Table 3 Optimal effectiveness of the two-stage algorithm and practical results

n	cast size	生产实绩			两阶段求解算法		
		makespan	JWT	MWT	makespan	JWT	MWT
49	$\{\{15, 11\}, \{10, 13\}\}$	2032	584	1055	1223	161	154
67	$\{\{15\}, \{3, 20, 20, 9\}\}$	2894	541	1164	2295	18	1887
79	$\{\{15, 21, 25\}, \{18\}\}$	3245	609	2320	2676	42	2239

根据表3可以得出以下结论: 1) 本文给出的两阶段优化算法对炉次等待设备时间和设备等待炉次时间的优化效果明显好于某钢铁企业的生产实绩. 加工时间可控的特性充分保证了减少炉次等待设备的时间, 即减少了因等待引起的热能损失. 2) 炉次等待设备时间的上下限约束使得JWT明显小于MWT, 即严格保证了炉次等待设备时间最小的优化目标. 3) 两阶段算法对makespan的优化效果要优于生产实绩. 因此, 本文给出的两阶段求解算法和模型适合应用于钢厂生产实际.

4.5 两阶段算法对钢厂计划优化效果分析 (Impact on performances of different waiting time weights)

基于不同工厂对炉次等待设备时间和设备等

待炉次时间的侧重不同, 本文设置炉次等待设备时间的权重 a 和设备等待炉次时间的权重 b 取3组数据, 分别为① $a = 0.3, b = 0.7$, ② $a = 0.6, b = 0.4$, ③ $a = 0.8, b = 0.2$. 实验中对生成的14个测试案例应用两阶段优化算法求解10次, 给出了每个实例10次求解结果的平均目标值、平均运行时间、两阶段最优解的目标值间比值的平均值以及等待时间比值的平均值, 分别表示为avgObj, avgCPU, avgPerO, avgPerW. 本节实验同时统计了应用GLS与数学规划方法相结合的求解算法在炉次等待设备和设备等待炉次不同权重组合下的求解性能. 设SSH表示应用SS与数学规划方法相结合的求解算法, GLSH表示应用GLS与数学规划方法相结合的求解算法. 统计结果如表4所示.

表4 不同等待时间权重对算法性能的影响

Table 4 Performances of the two-stage algorithm with different waiting time weights

n	m_1 * m_2 * m_3	算法	$a = 0.3, b = 0.7$				$a = 0.6, b = 0.4$				$a = 0.8, b = 0.2$			
			avgObj	avgCPU	avgPerW	avgPerO	avgObj	avgCPU	avgPerW	avgPerO	avgObj	avgCPU	avgPerW	avgPerO
45	2*2*2	SSH	823.12	34.83	0.2267	0.6552	790.66	36.53	0.1539	0.6460	768.66	35.07	0.1146	0.6437
		GLSH	923.72	20.79	0.3894	0.6622	881.92	24.49	0.2369	0.6143	836.56	34.83	0.1509	0.5643
	3*3*2	SSH	814.68	27.08	0.2182	0.6437	737.70	35.20	0.1293	0.6912	716.62	19.61	0.1032	0.7448
		GLSH	939.24	51.65	0.3453	0.6128	868.59	56.93	0.3695	0.6717	816.86	48.29	0.3976	0.7188
54	2*2*2	SSH	1155.68	111.05	0.2660	0.6580	1171.68	98.59	0.2077	0.5970	1088.24	119.32	0.1096	0.5483
		GLSH	1359.37	43.76	0.4287	0.7095	1292.68	55.55	0.2396	0.5754	1199.22	51.18	0.1338	0.4921
	3*3*2	SSH	1166.51	62.98	0.2731	0.6378	1048.1	50.76	0.2127	0.6662	1014.34	41.20	0.2746	0.7389
		GLSH	1338.59	121.99	0.3661	0.6053	1281.49	99.35	0.3779	0.6351	1156.04	98.58	0.4144	0.7008
54	2*2*2	SSH	1487.41	149.10	0.3338	0.6631	1384.54	175.17	0.2181	0.6115	1346.54	148.45	0.1387	0.5432
		GLSH	1650.69	55.36	0.3893	0.6658	1597.74	83.11	0.2883	0.5438	1406.36	96.24	0.1341	0.5085
	3*3*2	SSH	1370.42	100.56	0.2501	0.6353	1370.36	63.53	0.3319	0.6776	1254.42	61.61	0.3301	0.7353
		GLSH	1646.87	200.88	0.3896	0.6142	1583.50	147.73	0.3815	0.6273	1385.74	144.47	0.3729	0.6847
48	3*3*3	SSH	1042.74	149.10	0.3544	0.6357	1000.46	162.65	0.2376	0.5295	957.12	171.77	0.1363	0.4421
		GLSH	1201.01	87.43	0.4307	0.6594	1157.30	88.64	0.2853	0.5319	1086.20	74.61	0.1732	0.4375
	3*4*3	SSH	1035.34	163.33	0.3582	0.6552	1000.44	175.60	0.2466	0.5594	970.56	160.79	0.1593	0.4860
		GLSH	1186.91	136.20	0.4058	0.6218	1152.24	115.65	0.3209	0.5546	1076.90	95.15	0.2015	0.4606
	3*5*3	SSH	979.50	175.43	0.3300	0.6481	978.44	203.04	0.2531	0.5710	927.84	210.68	0.1688	0.5093
		GLSH	1172.34	140.41	0.4116	0.6330	1147.99	138.05	0.3175	0.5557	1067.20	131.79	0.2644	0.5265
	3*6*3	SSH	968.94	188.80	0.3346	0.6541	986.24	170.79	0.2517	0.5719	919.62	172.73	0.1603	0.5065
		GLSH	1186.03	121.93	0.4060	0.6282	1134.34	148.85	0.3291	0.5664	1033.90	126.69	0.2366	0.5074
66	3*3*3	SSH	1635.65	314.41	0.4789	0.6899	1472.98	363.67	0.2893	0.5529	1354.74	386.84	0.1680	0.4520
		GLSH	1746.83	201.24	0.4975	0.6899	1680.72	182.08	0.2884	0.5113	1521.98	230.75	0.1896	0.4415
	3*4*3	SSH	1504.04	377.73	0.3985	0.6453	1470.84	431.44	0.2772	0.5357	1345.7	404.64	0.1634	0.4417
		GLSH	1759.16	376.69	0.4580	0.6353	1662.12	280.72	0.3383	0.5460	1485.04	249.54	0.2252	0.4709
	3*5*3	SSH	1423.85	420.03	0.3633	0.6285	1474.9	437.08	0.2817	0.5337	1332.52	458.02	0.1911	0.4875
		GLSH	1744.41	300.92	0.4242	0.6101	1669.04	301.66	0.3242	0.5270	1460.89	284.06	0.2088	0.4516
	3*6*3	SSH	1415.94	463.52	0.3668	0.6428	1455.74	413.69	0.2735	0.5307	1314.42	447.02	0.1758	0.4695
		GLSH	1738.05	313.58	0.4515	0.6398	1657.86	322.15	0.3302	0.5422	1457.28	347.43	0.2327	0.4895

分析不同炉次等待时间、设备等待时间权重对SS和数学规划方法相结合求解算法性能的影响,根据表4可以得出以下结论:

1) 随着炉次等待设备时间权重的增加及设备等待炉次时间权重的减少,SSH求得的目标值、平均等待时间比值和平均目标值比值减少.当 $n = 32$,设备组合为 $2 * 2 * 2$ 时,等待时间权重 $a = 0.3$, $b = 0.7$ 的平均目标值、等待时间比值和目标时间比值明显比 $a = 0.6$, $b = 0.4$ 和 $a = 0.8$, $b = 0.2$ 的大.

2) SSH算法平均运行时间随炉次等待设备时间权重的增大而增大.当 $n = 66$,设备组合为 $3 * 3 * 3$ 时,SSH算法平均运行时间依 $a = 0.3$, $b = 0.7$, $a = 0.6$, $b = 0.4$ 和 $a = 0.8$, $b = 0.2$ 顺序增大.

3) 相同等待时间权重下,工件数相同的测试案例,随着设备数的增大平均目标值减少.充分说明了在给定的等待时间权重下,完成时间随着设备数的增大而减少.如 $n = 32$,设备组合分别为 $3 * 3 * 3$, $3 * 4 * 3$, $3 * 5 * 3$, $3 * 6 * 3$,等待时间权重 $a = 0.3$, $b = 0.7$ 时,平均完工时间随精炼设备数的增大而减少.

4) 相同等待时间权重下,SSH算法平均运行时间随工件数增大而增大.当 $a = 0.3$, $b = 0.7$ 及设备组合为 $3 * 3 * 3$ 时, $n = 48$ 的算法运行时间明显比 $n = 54$ 的运行时间小很多.

5) 相同等待时间权重和工件数下,SSH算法的平均运行时间随精炼设备的增多而增大,随精炼设备和炼钢设备的增多而减少.当 $a = 0.3$, $b = 0.7$, $n = 32$,设备组合 $2 * 2 * 2$ 的运行时间明显比 $3 * 3 * 2$ 大.当 $n = 48$ 时,算法的运行时间随设备组合 $3 * 3 * 3$, $3 * 4 * 3$ 中精炼设备的增大而增大.

对分别应用SS算法、GLS算法与CPLEX相结合的求解算法的优化结果进行分析,根据表4可以得出以下结论:

1) 应用SSH求得的最优目标值均优于应用GLSH算法求得的最优目标值.对于相同的设备等待时间和炉次等待时间权重组合,随着加工炉次数的增加,应用GLSH求得的最优目标值与SSH求得的最优值间的差距增大.同时,两者的差距依 $a = 0.3$, $b = 0.7$, $a = 0.6$, $b = 0.4$ 和 $a = 0.8$, $b = 0.2$ 顺序减少.

2) 应用GLSH求解的等待时间比值均大于应用SSH的等待时间比值,说明第一求解阶段应用本文给出的SS算法对设备等待时间和炉次等待时间的优化效果比GLS的优化效果好.

3) GLSH算法的运行时间优于SSH求解算法的时间,但两者的差距并不明显.基于表4给出的不同等待时间权重下,两阶段等待时间比值很小,目标值的比值依然很大,表示CPLEX求解阶段并没有因最小化最大完工时间,各工序加工时间取自区间内的最小值.因此,本文给出的两阶段求解算法充分保证了求得的调度计划的稳定性.通过对比分析第一求解阶段应用GLS和SS两种不同算法的求解性能,得出结论应用SSH的优化算法更适合求解本文研究的问题.

5 结论(Conclusions)

炼钢-精炼-连铸是钢铁生产中的重要环节,为实现炼钢、连铸、热轧一体化生产起着重要的协调作用.本文根据实际生产中加工时间可控的特点,设计了SS算法和数学规划相结合的两阶段求解算法.为了测试本文给出的两阶段算法和模型的优化效果,根据某钢铁企业生产实际随机生成了14个案例进行实验,实验结果表明:

1) 基于加工时间可控的特性,数学规划求解阶段对等待时间的优化结果明显比SS算法的优化结果好.

2) 两阶段优化算法求得的最优值、算法运行时间与炉次等待设备时间的权重成反比的关系.

3) 两阶段求解算法对makespan和等待时间的优化效果明显好于钢厂的生产实绩.

4) 第一求解阶段应用SS的混合求解算法的优化效果优于第1阶段应用GLS的混合算法的求解结果.

参考文献(References):

- [1] TANG L X, LIU J Y, RONG A Y, et al. A review of planning and scheduling and methods for integrated steel production [J]. *European Journal of Operational Research*, 2001, 133(1): 1 - 20.
- [2] TANG L X, LIU J Y, RONG A Y, et al. A mathematical programming model for scheduling steelmaking-continuous casting production [J]. *European Journal of Operational Research*, 2000, 120(2): 423 - 435.
- [3] TANG L X, LUH P B, LIU J Y, et al. Steel-making process scheduling using lagrangian relaxation [J]. *International Journal of Production Research*, 2002, 40(1): 55 - 70.
- [4] BELLABDAOUI A, TEGHEM J. A mixed-integer linear programming model for the continuous casting planning [J]. *International Journal of Production Economics*, 2006, 104(2): 260 - 270.
- [5] ATIGHEHCHIAN A, BIJARI M, TARKESH H. A novel hybrid algorithm for scheduling steel-making continuous casting production [J]. *Computer & Operation Research*, 2009, 36(8): 2450 - 2461.
- [6] 李铁克, 苏志雄. 炼钢连铸生产调度问题的两阶段遗传算法 [J]. 中国管理科学, 2009, 15(5): 68 - 74.
(LI Tiek, SU Zhixiong. Two-stage genetic algorithm for SM CC production scheduling [J]. *Chinese Journal of Management Science*, 2009, 15(5): 68 - 74.)

- [7] TANG L X, GUAN J, HU G F. Steelmaking and refining coordinated scheduling problem with waiting time and transportation consideration [J]. *Computer & Industrial Engineering*, 2010, 58(2): 239 – 248.
- [8] GUPTA J N D, KRUGER K, LAUFF V, et al. Heuristics for hybrid flow shops with controllable processing times and assignable due dates [J]. *Computer & Operations Research*, 2002, 29(10): 1417 – 1439.
- [9] 俞胜平, 庞新富, 柴天佑, 等. 加工时间不确定的炼钢连铸生产调度问题的研究 [J]. *控制与决策*, 2009, 24(10): 1467 – 1472.
(YU Shengping, PANG Xinfu, CHAI Tianyou, et al. Research on production scheduling for steelmaking and continuous casting with processing time uncertainty [J]. *Control and Decision*, 2009, 24(10): 1467 – 1472.)
- [10] GLOVER F. Heuristics for integer programming using surrogate constraints [J]. *Decision Sciences*, 1977, 8(1): 156 – 166.
- [11] RUSSELL R A, CHIANG W C. Scatter search for the vehicle routing problem with windows [J]. *European Journal of Operation Research*, 2006, 169(2): 606 – 622.
- [12] SARAVANAN M, HAQ A N, VIVEKRAJ A R, et al. Performance evaluation of the scatter search method for permutation flowshop sequencing problems [J]. *International Journal of Advanced Manufacturing Technology*, 2008, 37(11/12): 1200 – 1208.
- [13] TANG J F, ZHANG J, PAN Z D. A scatter algorithm for solving vehicle routing problem with loading cost [J]. *Expert System with Applications*, 2010, 37(6): 4073 – 4083.
- [14] NOORUL H A, SARAVANAN M, VIVEKRAJ A R, et al. A scatter search approach for general flowshop scheduling problem [J]. *International Journal of Advanced Manufacturing Technology*, 2007, 31(7/8): 731 – 736.
- [15] NOWICKI E, SMUTNICKI C. The flow shop with parallel machines: a tabu search approach [J]. *European Journal of Operation Research*, 1998, 106(2/3): 226 – 253.

作者简介:

谭园园 (1983–), 女, 博士研究生, 研究方向为生产计划与调度、炼钢连铸热轧一体化计划与调度, E-mail: tanyuanyuan83@sina.cn;

宋健海 (1965–), 男, 博士, 教授级高级工程师, 研究方向为钢铁企业ERP、MES及企业信息系统整体解决方案等, E-mail: songjianhai@baosight.com;

刘士新 (1968–), 男, 教授, 博士生导师, 研究方向为生产计划与调度、物流建模与优化, 智能优化算法, E-mail: sxliu@mail.neu.edu.cn.