

# 求解炼钢-连铸生产调度问题的拉格朗日算法

毛 坤, 潘全科, 庞新富

(东北大学流程工业综合自动化国家重点实验室, 辽宁 沈阳 110819)

**摘要:** 炼钢-连铸是现代钢铁企业生产流程的核心工序. 本文将炼钢-连铸调度问题抽象为具有复杂约束的混合流水车间调度问题. 以炉次总驻留时间最短和浇次准时开浇为优化目标, 以炉次连浇为等式约束, 建立了该问题的0-1型混合整数非线性模型. 通过松弛资源析取约束, 解除连续变量和整数变量之间的耦合关系, 将松弛问题分解成两个简单的子问题. 提出一种拉格朗日松弛水平算法, 采用对偶松弛函数下降水平控制策略, 以保证算法收敛于对偶松弛问题最优解. 不同的测试算例表明: 与传统拉格朗日松弛算法相比, 所提算法可得到更好的下界和对偶间隙, 并且求解时间较短. 尤其问题规模越大, 其优越性更加明显.

**关键词:** 炼钢-连铸; 生产调度; 拉格朗日松弛水平算法; 线性规划

**中图分类号:** TP273      **文献标识码:** A      **文章编号:** 1000-5781(2014)02-0233-13

## Lagrangian algorithm for steelmaking-continuous casting production scheduling

Mao Kun, Pan Quanke, Pang Xinfu

(State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China)

**Abstract:** Steelmaking-continuous casting is the core production process in modern steel production. It is abstracted as a hybrid flowshop with complex constraints in this paper. A 0-1 mixed-integer nonlinear programming model, which treats the no dead time within the same cast at the last stage as an equality constraint and uses the total sojourn times of all jobs and the penalty of the earliness and tardiness of all casts as its minimum objective, is established. The disjunctive constraints of this model are relaxed in order to decouple the coupling relationship between continuous variables and integer variables. So the relaxed problem can be decomposed into two tractable sub-problems. A Lagrangian relaxation(LR) level algorithm is proposed for this scheduling problem. The algorithm applies the decreasing level control approach of the dual relaxation function evaluation in order to guarantee convergence. A computational comparison is provided for the LR level algorithm and traditional LR algorithm. Numerical results show that the LR level algorithm method can provide better lower bound, better dual gap and faster convergence for medium sized problems, especially for larger sized problems.

**Key words:** steelmaking-continuous casting; production scheduling; Lagrangian relaxation level algorithm; linear programming

收稿日期: 2012-06-07; 修订日期: 2013-01-21.

基金项目: 高等学科创新与引智计划资助项目(B08015); 教育部项目基本科研业务费资助项目(N110208001); 国家自然科学基金资助项目(61174187; 71021061; 60974091; 61104174); 中央高校基本科研业务费资助项目(N100508001); 东北大学科研启动经费资助项目(29321006).

## 1 引言

炼钢-连铸是现代钢铁企业生产流程的核心工序,涉及炼钢、精炼和连铸三道工序的一体化作业.科学地调度炼钢-连铸生产过程可以充分提高企业生产率和准时交货率,降低物耗和能耗,从而增强市场竞争力.

许多学者从不同的角度对炼钢-连铸生产调度问题进行了研究<sup>[1-11]</sup>,所建立的问题模型主要有三种:基于最小时间单元的纯整数模型<sup>[1-3]</sup>;基于析取图的图论模型<sup>[4]</sup>;基于0-1变量的混合整数模型<sup>[5,6]</sup>.模型求解方法可分为四类:基于拉格朗日松弛框架的动态规划方法<sup>[1-3]</sup>;整数规划和分支定界方法<sup>[5]</sup>;智能算法(或启发式方法)<sup>[6-10]</sup>;图论方法<sup>[11]</sup>.上述方法中,整数规划和分支定界方法一般时间复杂度很高,可以解决小规模问题,但随着问题规模增大,计算时间呈指数级增长;基于拉格朗日松弛框架的动态规划方法求解质量很好,但时间复杂度较高<sup>[1-3]</sup>,且由于无法预先知道对偶松弛问题的最优解,难以保证算法收敛<sup>[12-15]</sup>;基于析取图论的束搜索方法<sup>[11]</sup>,适用于小规模问题,但问题复杂时难以确定最佳搜索路径;智能算法<sup>[6-10]</sup>计算效率较高,但是如果第一阶段产生的粗调度不够理想,第二阶段可以优化的范围就非常有限,另外该类算法无法保证其迭代的收敛性.

基于已有研究,以炉次总驻留时间最短和浇次准时开浇为优化目标,建立了炼钢-连铸生产调度的0-1型混合整数非线性模型.该模型通过对炉次驻留时间进行惩罚,避免因炉次停留时间过长而降低冶金性能;通过对炉次提前/拖期进行惩罚,以保证准时生产.在求解时,将目标函数中的非线性的目标函数转化成线性的目标函数.通过松弛资源析取约束,解除连续变量和整数变量之间的耦合关系,将松弛问题分解成两个独立的容易求解的子问题.提出一种拉格朗日松弛水平算法,采用对偶松弛函数下降水平控制策略,以保证其收敛于对偶松弛问题最优解,并给出了收敛性证明.另外,针对算法的函数值震荡下降逼近最优值的特点,提出了新的可行化构造方法,以保证得到较优的原问题可行解.

## 2 炼钢-连铸生产调度模型

### 2.1 炼钢-连铸生产过程

钢铁企业的炼钢-连铸生产过程包含炼钢、精炼和连铸三个阶段,其中每个生产阶段可能存在多个并行的生产设备.炼钢-连铸的主要生产过程如下:炼钢-连铸区接受来自高炉的铁水,通过炼钢阶段的转炉设备将冶炼好的铁水转换为钢水,然后将钢水倒入钢包,并载运到精炼设备进行精炼,进一步调整钢水的温度和成分.精炼后的钢水被运送到指定的连铸机前倒入中间包,并经中间包流入连铸机连续铸成板坯.其中,一个运送钢水的钢包被称为炉次,也称“工件”,是炼钢-连铸最小生产单元;在同一台连铸机上连续浇铸的炉次集合称为一个浇次,是炼钢-连铸的最大生产单元.

例如,在某大型钢厂的炼钢-连铸生产过程中,炼钢阶段有3台转炉(LD1, LD2, LD3),精炼阶段有3台RH设备(RH1, RH2, RH3),连铸阶段有3台连铸机(CC1, CC2, CC3),三个炉次的加工路径分别为:1LD → 1RH → 2CC, 2LD → 2RH → 1CC, 3LD → 3RH → 3CC.图1给出了炼钢-连铸生产过程示意甘特图.浇次1(cast)中有三个炉次(charge),每个炉次的加工路径分别为炉次1(1LD → 2RH → 3CC),炉次2(2LD → 1RH → 3CC),炉次3(1LD → 1RH → 3CC).

炼钢-连铸生产过程有如下特点:

1) 所有炉次的加工顺序相同,即每个炉次都要经过冶炼、精炼和连铸三个阶段;每阶段都有多台并行生产设备;同一台连铸机必须在一定炉次范围内连续浇注,即每个浇次中的炉次必须连续加工;

2) 每台连铸机上所加工的浇次预先给定;各连铸机上的相邻浇次之间需要一定的间隔时间,以更换结晶器和调整设备;

3) 生产过程为高温连续作业,等待时间的增加可能意味着温度的降低,再升温将会导致能源和生产成本

的增加.

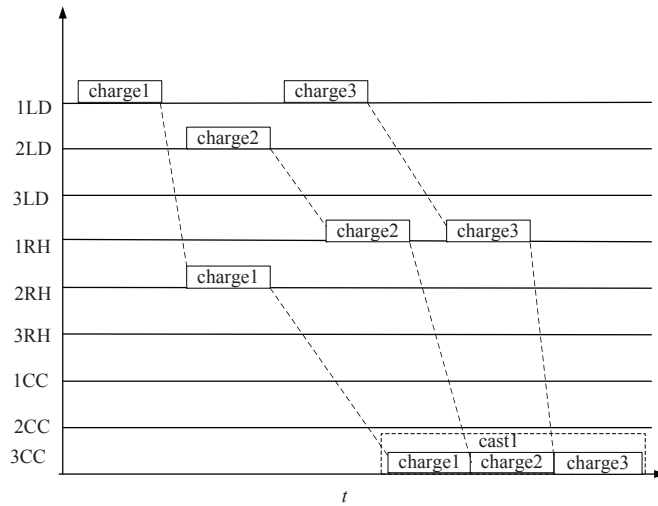


图 1 炼钢-连铸生产甘特图

Fig. 1 The Gantt chart for steelmaking-continuous casting process

### 2.2 炼钢-连铸生产调度问题

本文研究的炼钢-连铸调度问题, 生产流程包括炼钢、精炼和连铸三阶段, 各阶段又存在着多台并行生产设备, 每个炉次的加工顺序都相同, 且每个浇次中的所有炉次在连铸机上需要连续加工. 因而可以将其抽象为最后阶段连续加工的混合流水车间调度(hybrid flow-shop, HFS)问题. 炼钢-连铸生产调度就是在一定的设备条件下, 将浇次计划中的炉次按照规定的生产工艺合理地分配到设备上, 并确定其开始加工时间, 使得各浇次准时开浇以及各炉次的总驻留时间最小. 其中浇次所包含的炉次、浇次所分配的连铸机、各浇次的浇注次序、各浇次的计划开浇时间、炉次的加工时间和炉次的运输时间均为已知, 炉次在炼钢和精炼阶段上的设备指派不受限制.

### 2.3 符号定义

为便于描述, 定义符号和变量如下:

#### 1) 索引、元素和集合

$i, r$  为炉次序号,  $\Omega$  为含有所有炉次的集合,  $i, r \in \Omega, |\Omega|$  为总炉次数;  $N$  表示浇次总数;  $n$  为浇次序号,  $n = 1, 2, \dots, N$ ;  $j$  为阶段序号, 冶炼、精炼和连铸三个阶段分别对应为  $j = 1, 2, 3$ ;  $M_j$  表示第  $j$  阶段的设备数(整数),  $M_j \geq 1$ ;  $\Omega_n$  表示第  $n$  个浇次中所有炉次的有序集合,  $|\Omega_n|$  表示第  $n$  个浇次中炉次的个数, 且有  $\Omega_{n_1} \cap \Omega_{n_2} = \emptyset, \forall n_1, n_2 = 1, 2, \dots, N, n_1 \neq n_2$ , 且  $\Omega_1 \cup \Omega_2 \cup \dots \cup \Omega_n = \Omega$ ;  $s(n)$  表示第  $n$  个浇次的末炉次序号,  $s(n) = s(n-1) + |\Omega_n|, s(0) = 0, n = 1, 2, \dots, N$ , 则  $\Omega_n = \{s(n-1) + 1, \dots, s(n)\}$ ;  $B_k$  表示第  $k$  个连铸机上已指派的浇次的有序集合,  $|B_k|$  表示第  $k$  个连铸机上浇次的个数,  $1 \leq k \leq M_3$ ;  $b(k)$  为第  $k$  台连铸机上的末浇次序号,  $b(k) = b(k-1) + |B_k|, b(0) = 0, k = 1, 2, \dots, M_3$ , 则  $B_k = \{b(k-1) + 1, \dots, b(k)\}$ .

#### 2) 固定参数

$P_{ij}$  表示炉次  $i$  在第  $j$  阶段的作业时间;  $T_{j,j+1}$  表示第  $j$  阶段和第  $j+1$  阶段之间的标准运输时间;  $d_n$  表示浇次  $\Omega_n$  的计划开浇时间;  $T_B$  表示相邻浇次间因更换结晶器所需最小间隔时间;  $C_1$  表示炉次  $i$  的总驻留时间惩罚;  $C_2$  为开浇提前惩罚系数;  $C_3$  开浇延迟惩罚系数;  $U$  为足够大的正数.

#### 3) 决策变量

$x_{ijk}$  为 0-1 变量, 当且仅当炉次  $i$  被指派到阶段  $j$  的第  $k$  台设备上时其为 1, 否则为零;  $y_{irj}$  为 0-1 变量, 当且仅当在炉次  $i$  先于  $r$  在阶段  $j$  被处理时为 1, 否则为零;  $t_{ij}$  为炉次  $i$  在第  $j$  阶段的作业开始时间.

## 2.4 模型建立

### 2.4.1 目标函数

炼钢-连铸生产调度的目标是炉次的总驻留时间最短和浇次的准时开浇, 即

$$\text{Min } G = F_1 + F_2, \quad (1)$$

其具体含义如下.

1) 所有炉次的总驻留时间惩罚函数

$$F_1 = C_1 \sum_{i=1}^{|\Omega|} (t_{i,3} - t_{i,1}); \quad (2)$$

2) 浇次开浇时间提前或拖期惩罚函数

$$F_2 = C_2 \sum_{n=1}^N \max(0, d_n - t_{s(n-1)+1,3}) + C_3 \sum_{n=1}^N \max(0, t_{s(n-1)+1,3} - d_n), \quad (3)$$

其中  $F_2$  的第一部分为浇次开浇时间提前惩罚函数,  $F_2$  的第二部分为浇次开浇时间拖期惩罚函数.

### 2.4.2 连续变量相关约束

1) 炉次加工顺序约束. 同一炉次在前一阶段加工完毕并运达下一阶段后, 才能开始加工, 即

$$t_{i,j+1} - t_{ij} - P_{ij} \geq T_{j,j+1}, \forall i \in \Omega, \forall j = 1, 2; \quad (4)$$

2) 连浇约束. 在同一连铸机上, 浇次内的相邻炉次必须连续加工, 即

$$t_{i+1,3} = t_{i,3} + P_{i3}, \forall i, i+1 \in \Omega_n, n = 1, 2, \dots, N; \quad (5)$$

3) 浇次间隔约束. 同一台连铸机上的相邻浇次之间需要一定的间隔时间, 以便更换相关设备, 如结晶器等, 即

$$t_{i+1,3} - t_{i,3} - P_{i3} \geq T_B, \quad (6)$$

其中  $i = s(b(k-1) + n), \forall k = 1, 2, \dots, M_3, \forall n = 1, 2, \dots, |B_k| - 1$ ;

4) 时间变量约束. 时间变量为非负, 即

$$t_{ij} \geq 0, \forall i \in \Omega, \forall j = 1, 2, 3. \quad (7)$$

### 2.4.3 整数变量相关约束

1) 炉次在各阶段加工设备唯一性约束

$$\sum_{k=1}^{M_j} x_{ijk} = 1, \forall i \in \Omega, \forall j = 1, 2; \quad (8)$$

2) 顺序关系约束

$$y_{irj} + y_{rij} = 1, \forall i \neq r, i \in \Omega, r \in \Omega, \forall j = 1, 2; \quad (9)$$

3) 整数变量约束.

$$x_{ijk} \in \{0, 1\}, \forall i \in \Omega, \forall j = 1, 2, \forall k = 1, 2, \dots, M_j, \quad (10)$$

$$y_{irj} \in \{0, 1\}, \forall i \neq r, i \in \Omega, r \in \Omega, \forall j = 1, 2. \quad (11)$$

### 2.4.4 整数变量与连续变量耦合约束

设备能力的析取约束. 在同一设备(转炉或精炼炉)不能同时加工多个炉次, 即

$$t_{rj} - t_{ij} - P_{ij} + (3 - x_{ijk} - x_{rjk} - y_{irj})U \geq 0, \forall i \neq r, i \in \Omega, r \in \Omega, \forall j = 1, 2. \quad (12)$$

### 3 模型转换

由于目标函数  $F_2$  为简单的非线性函数,可换将其转换成线性函数<sup>[16]</sup>,令

$$t_{s(n-1)+1,3}^l = \max\left(0, d_n - t_{s(n-1)+1,3}\right), \quad t_{s(n-1)+1,3}^u = \max\left(0, t_{s(n-1)+1,3} - d_n\right),$$

可知  $t_{s(n-1)+1,3} = t_{s(n-1)+1,3}^u - t_{s(n-1)+1,3}^l + d_n$ . 因此  $F_2$  可转换成如下形式

$$F_2 = C_2 \sum_{n=1}^N t_{s(n-1)+1,3}^l + C_3 \sum_{n=1}^N t_{s(n-1)+1,3}^u. \quad (13)$$

上述目标函数在满足约束条件式(4)–式(12)以外,还满足如下条件

$$t_{s(n-1)+1,3} = t_{s(n-1)+1,3}^u - t_{s(n-1)+1,3}^l + d_n, \forall n = 1, 2, \dots, N, \quad (14)$$

$$t_{s(n-1)+1,3}^u \geq 0, \quad t_{s(n-1)+1,3}^l \geq 0, \forall n = 1, 2, \dots, N. \quad (15)$$

与文献[16]方法不同,本文没有将转换后的  $t_{s(n-1)+1,3}^u - t_{s(n-1)+1,3}^l + d_n$  替换原问题中的  $t_{i,3}$ . 这样模型表述更简洁,求解更简单.

### 4 模型求解

#### 4.1 拉格朗日松弛

通过松弛资源析取约束式(12),解除连续变量和整数变量之间的耦合关系,将松弛问题划分成两个独立的容易求解的子问题,即只含连续变量的线性规划问题和只含整数变量的简单整数规划问题.

因此引入非负的拉格朗日乘子  $\mu$  将资源析取约束松弛到目标函数,有

$$(LR) \begin{cases} \text{Min } L(\mu) = G_L(\mathbf{x}, \mathbf{y}, \mathbf{t}, \mu) = \text{Min}\{F_1 + F_2 + F_3\} \\ \text{s.t. 式(4) – 式(11), 式(14), 式(15),} \end{cases}$$

其中  $F_1$  和  $F_2$  的定义与式(2)和式(13)相同,  $F_3 = -\sum_{j=1}^2 \sum_{k=1}^{M_j} \sum_{i=1}^{|\Omega|} \sum_{r=1, r \neq i}^{|\Omega|} \mu_{irjk} \bar{F}_{irjk}$ ,  $\bar{F}_{irjk} = t_{rj} - t_{ij} - P_{ij} + (3 - x_{ijk} - x_{rjk} - y_{irj})U$ ,  $\mathbf{x} = (x_{ijk})_{N_x \times 1}$  是一个  $N_x \times 1$  的向量,  $\mathbf{y} = (y_{irj})_{N_y \times 1}$ ,  $\mathbf{t} = (t_{ij})_{N_t \times 1}$ ,  $\mu = (\mu_{irjk})_{N_\mu \times 1}$ ,  $N_x = |\Omega| \left( \sum_{j=1}^2 M_j \right)$ ,  $N_y = 2|\Omega|(|\Omega| - 1)$ ,  $N_t = 2|\Omega|$ ,  $N_\mu = |\Omega|(|\Omega| - 1) \left( \sum_{j=1}^2 M_j \right)$ .

因此其拉格朗日对偶问题为

$$(LD) \begin{cases} \text{Max } L(\mu) = \text{Max Min } G_L \\ \text{s.t.} \\ \text{式(4) – 式(11), 式(14), 式(15)} \\ \mu \geq 0. \end{cases} \quad (16)$$

对于给定的  $\mu$ , 松弛之后的目标函数  $G_L$  可分解成两部分,一部分为只含连续变量的目标函数  $G_{L1}$ , 另一部分为只含整数变量的目标函数  $G_{L2}$ , 即

$$G_{L1} = F_1 + F_2 + F_{31}, \quad G_{L2} = F_{32}, \quad (17)$$

其中  $F_3 = F_{31} + F_{32}$ ,  $F_{31} = -\sum_{j=1}^2 \sum_{k=1}^{M_j} \sum_{i=1}^{|\Omega|} \sum_{r=1, r \neq i}^{|\Omega|} \mu_{irjk} (t_{rj} - t_{ij} - P_{ij})$ ,  $F_{32} = -\sum_{j=1}^2 \sum_{k=1}^{M_j} \sum_{i=1}^{|\Omega|} \sum_{r=1, r \neq i}^{|\Omega|} a_{irjk} \mu_{irjk} U$ ,

$$a_{irjk} = 3 - x_{ijk} - x_{rjk} - y_{irj}.$$

经过松弛,模型约束是可分离的.将问题(LR)解耦成两个简单子问题来求解,即只有连续变量的子问题(LR<sub>1</sub>)和只有整数变量的子问题(LR<sub>2</sub>).

$$(LR_1) \begin{cases} \text{Min } G_{L1}(\mathbf{t}), \\ \text{s.t. 式(4) - 式(7), 式(14), 式(15),} \end{cases} \quad (18)$$

$$(LR_2) \begin{cases} \text{Min } G_{L2}(\mathbf{x}, \mathbf{y}), \\ \text{s.t. 式(8) - 式(11).} \end{cases} \quad (19)$$

## 4.2 对偶松弛问题求解

### 4.2.1 连续变量子问题求解

子问题(LR<sub>1</sub>)是简单的线性规划问题.对于给定的拉格朗日乘子 $\mu$ ,问题(LR<sub>1</sub>)的目标函数 $G_{L1}$ 和约束都是线性函数,利用内点算法或单纯形法即可求解.

### 4.2.2 整数变量子问题求解

子问题(LR<sub>2</sub>)是简单的整数规划问题.对于给定的拉格朗日乘子 $\mu$ ,问题(LR<sub>2</sub>)的目标函数 $G_{L2}$ 中的指派变量 $\mathbf{x}$ 和顺序变量 $\mathbf{y}$ 是可分离的,即 $G_{L2}$ 可分解成两个部份

$$G_{L2} = G_{L2}(\mathbf{x}) + G_{L2}(\mathbf{y}), \quad (20)$$

$$\text{其中 } G_{L2}(\mathbf{x}) = \sum_{j=1}^2 \sum_{k=1}^{M_j} \sum_{i=1}^{|\Omega|} \left( \sum_{r=1}^{|\Omega|} \mu_{irjk} + \sum_{r=1}^{|\Omega|} \mu_{rijk} \right) x_{ijk} U, \quad G_{L2}(\mathbf{y}) = \sum_{j=1}^2 \sum_{i=1}^{|\Omega|} \sum_{r=1, r \neq i}^{|\Omega|} \sum_{k=1}^{M_j} (\mu_{irjk} y_{irj} - 3) U.$$

问题(LR<sub>2</sub>)的约束中两者无耦合,其约束较为简单.问题(LR<sub>2</sub>)可以分解成分别关于指派变量 $\mathbf{x}$ 和顺序变量 $\mathbf{y}$ 的两个子问题,即

$$(LR_{21}) \begin{cases} \text{Min } G_{L2}(\mathbf{x}) \\ \text{s.t. 式(8), 式(10),} \end{cases} \quad (21)$$

$$(LR_{22}) \begin{cases} \text{Min } G_{L2}(\mathbf{y}) \\ \text{s.t. 式(9), 式(11).} \end{cases} \quad (22)$$

对于给定的拉格朗日乘子, $G_{L2}(U)$ 是常数.由于约束(8)和(10)是简单约束,易知问题(LR<sub>21</sub>)中的指派变量的最优点为

$$x_{ijk}^* = \begin{cases} 1, & k = \arg \min_{1 \leq k \leq M_j} \bar{\mu}_{ijk} \\ 0, & \text{其它,} \end{cases} \quad \bar{\mu}_{ijk} = \sum_{r=1}^{|\Omega|} \mu_{irjk} + \sum_{r=1}^{|\Omega|} \mu_{rijk}, \quad i = 1, 2, \dots, |\Omega|, j = 1, 2,$$

问题(LR<sub>22</sub>)中的顺序变量的最优点为

$$y_{irj}^* = \begin{cases} 1, & \hat{\mu}_{ijk} < \hat{\mu}_{rjk} \\ 0, & \text{其它,} \end{cases} \quad \hat{\mu}_{ijk} = \sum_{r=1, r \neq i}^{|\Omega|} \sum_{k=1}^{M_j} \mu_{irjk} y_{irj}, \quad r \neq i, i, r = 1, 2, \dots, |\Omega|, j = 1, 2.$$

## 4.3 拉格朗日松弛水平算法

传统拉格朗日松弛方法如要收敛,则需事先知道对偶松弛问题的最优值,但实际上很难找到这类对偶问题的最优值<sup>[17]</sup>;在实际求解中,一般采用迭代的最好原问题可行解作为对偶松弛问题的最优值,以得到相对次梯度的步长,但此方法难以保证算法的收敛性<sup>[17]</sup>.针对以上不足,本文提出一种拉格朗日松弛水

平算法,在不知道对偶松弛问题的最优值的情况下,通过不断估计目标函数值的办法逐渐逼近问题最优值<sup>[18,19]</sup>,最终达到收敛.同时,与文献[1-3]的拉格朗日松弛算法不同,所提算法不需要在每次迭代过程中都求解原问题可行解,因而能大幅度提高求解效率.

令  $F(\boldsymbol{\mu}) = -L(\boldsymbol{\mu})$ ,  $v_{irjk} = t_{rj} - t_{ij} - P_{ij} + (3 - x_{ijk} - x_{rjk} - y_{irj})U$ ,  $h(l)$  是关于  $l$  的一个已知函数.拉格朗日松弛水平算法的步骤如下:

**步骤 1** (初始化) 设初始值  $\boldsymbol{\mu}_0 \geq 0$ ,  $\varepsilon_1 > 0$ ,  $\varepsilon_2 > 0$ ,  $\varepsilon_3 > 0$ ,  $\beta \in (0, 1)$ ,  $W > 0$  (为偶数),  $r \leftarrow 0$  (记录  $F(\boldsymbol{\mu}_m)$  循环跳跃次数),  $r_1 \leftarrow 0$ ,  $r_2 \leftarrow 0$ ,  $l \leftarrow 1$ ,  $t \in (0, 2)$ ,  $\delta_1 > 0$  和  $R > 0$ .  $\sigma_1 \leftarrow 0$ ,  $F_{\text{rec}}^0 \leftarrow \infty$ ,  $m \leftarrow 1$ ,  $M[l] = 1$  ( $M[l]$  记录  $F_{\text{lev}}^m$  的第  $m$  次改变的迭代次数);

**步骤 2** (求解对偶松弛目标函数值) 计算  $F(\boldsymbol{\mu}_m)$  和次梯度  $\mathbf{g}_F(\boldsymbol{\mu}_m) = (v_{irjk})_{N_{\mu} \times 1}$  (简记为  $\mathbf{g}_m$ ).

如果  $F(\boldsymbol{\mu}_m) < F_{\text{rec}}^{m-1}$ , 则  $F_{\text{rec}}^m \leftarrow F(\boldsymbol{\mu}_m)$ ,  $\boldsymbol{\mu}_{\text{rec}}^m \leftarrow \boldsymbol{\mu}_m$ ; 否则  $F_{\text{rec}}^m \leftarrow F_{\text{rec}}^{m-1}$ ,  $\boldsymbol{\mu}_{\text{rec}}^m \leftarrow \boldsymbol{\mu}_{\text{rec}}^{m-1}$ . 这样可以保证  $F_{\text{rec}}^m = \min \{F(\boldsymbol{\mu}_j) \mid 0 \leq j \leq m\}$ ;

**步骤 3** (充分下降检测) 如果  $F(\boldsymbol{\mu}_m) \leq F_{\text{rec}}^m - 0.5\delta_l$ , 则  $M(l+1) \leftarrow m$ ,  $\sigma_m \leftarrow 0$ ,  $\delta_{l+1} \leftarrow \delta_l$ ,  $l \leftarrow l+1$ , 转到步骤 5;

**步骤 4** (微振荡检测)  $D[r] \leftarrow F(\boldsymbol{\mu}_m)$ , 如果  $r > 2$  且  $D[r] = D[1]$ , 则  $r_1 \leftarrow r_1 + 1$ ; 如果  $r > 2$  且  $D[r] = D[2]$ , 则  $r_2 \leftarrow r_2 + 1$ ,  $r \leftarrow r + 1$ . 如果  $r > W$ , 则  $r \leftarrow 1$ ,  $r_1 \leftarrow 0$ ,  $r_2 \leftarrow 0$ ; 如果  $r_1 = r_2 = 0.5W$ , 则  $\delta_m \leftarrow R + 1$ ,  $r \leftarrow 1$ ,  $r_1 \leftarrow 0$ ,  $r_2 \leftarrow 0$ ;

**步骤 5** (粗振荡检测) 如果  $\delta_m > h(l)R$ , 则  $M(l+1) \leftarrow m$ ,  $\sigma_m \leftarrow 0$ ,  $\delta_{l+1} \leftarrow \beta\delta_l$  且  $\boldsymbol{\mu}_m \leftarrow \boldsymbol{\mu}_{\text{rec}}^m$ ,  $\mathbf{g}_m \leftarrow \mathbf{g}_F(\boldsymbol{\mu}_{\text{rec}}^m)$ ,  $l \leftarrow l + 1$ ;

**步骤 6** (投影) 设  $F_{\text{lev}}^m = F_{\text{rec}}^{m(l)} - \delta_l$ , 令

$$\mathbf{z}_m = \boldsymbol{\mu}_m - \mathbf{d}_m, \boldsymbol{\mu}_{m+1} = P_{R^+}(\mathbf{z}_m), \mathbf{d}_m = t(F(\boldsymbol{\mu}_m) - F_{\text{lev}}^m)\mathbf{g}_m / \|\mathbf{g}_m\|^2,$$

其中  $P_{R^+}(\mathbf{z}_m)$  将  $\mathbf{z}_m$  投影到非负区域上;

**步骤 7** (路径更新)  $\sigma_{m+1} \leftarrow \sigma_m + \|\mathbf{d}_m\|$ ,  $m \leftarrow m + 1$ ;

**步骤 8** (停止条件) 如果  $\|\mathbf{d}_m\| < \varepsilon_1$ ,  $\|\boldsymbol{\mu}_m\| < \varepsilon_2$  和  $|\delta_l / F_{\text{lev}}^{m(l)}| < \varepsilon_3$  中任一个成立, 则停止; 否则转到步骤 2;

**步骤 9** (构造最优可行解) 根据已得到的对偶松弛问题解, 利用 4.4 和 4.5 节的构造相对最优可行解的方法可得到一个相对最优可行解.

#### 4.4 构造可行解

一般来说, 拉格朗日松弛问题解相对于原问题是不可行的, 需要在松弛解的基础上构造原问题可行解. 由于松弛了资源析取约束, 不同炉次在同一台设备上加工有可能冲突, 其解不可行主要原因是, 同一台设备上的所有炉次加工顺序之间可能出现有向环. 例如松弛解可能会出现如下情况: 在第 1 阶段的第 1 台机器上的 3 个加工炉次 (炉次 1、2、3), 其指派变量的解为  $x_{111} = 1$ 、 $x_{211} = 1$  和  $x_{311} = 1$ , 顺序变量的解为  $y_{121} = 1$  (炉次 1 先于 2 加工)、 $y_{231} = 1$  (炉次 2 先于 3 加工) 和  $y_{311} = 1$  (炉次 3 先于炉次 1 加工). 此情况下, 加工顺序明显矛盾, 因为根据前面的结果 ( $y_{121} = 1$  和  $y_{231} = 1$ ), 炉次 1 应先于 3 加工, 即  $y_{311} = 1$ .

为了消除有向环, 可通过每个炉次的序变量  $y_{irj}$  之和的大小确定所有的炉次的加工顺序, 然后将炉次均衡地指派到各个加工机器上, 从而确定了每个阶段每个炉次的加工设备, 即确定了指派变量  $x_{ijk}$ . 这样原问题的整数变量都已确定, 其简化为一个线性规划问题, 可利用单纯形法或内点法求解, 从而得到各个炉次在各个阶段的开始加工时间  $t_{ij}$ . 具体步骤如下:

**步骤 1** 求解松弛问题, 确定顺序变量  $y_{irj}$ ;

**步骤 2** 修正部分顺序变量  $y_{irj}$ , 其修正方式如下: 如果  $i < r$ , 则  $y_{irj} \leftarrow 1$ ; 否则,  $y_{irj} \leftarrow 0$ ,  $i, r \in B_k$ ,  $1 \leq k \leq M_3$ ,  $j = 1, 2$ ;

**步骤 3** 求每个炉次的序变量之和, 即  $s_{ij} = \sum_{r=1}^{|\Omega|} y_{irj}, 1 \leq r \leq |\Omega|, j = 1, 2;$

**步骤 4** 炉次的序变量之和  $s_{ij}$ , 按照从小到大的顺序排序, 得到每个阶段的所有炉次的加工顺序  $(i_{1,j}, i_{2,j}, \dots, i_{|\Omega|,j}), j = 1, 2.$  即确定如下关系:  $y_{ir,ir+m,j} = 1, y_{ir+m,ir,j} = 0, 1 \leq r \leq |\Omega| - 1, 1 \leq m \leq |\Omega| - r - 1, j = 1, 2;$

**步骤 5** 根据炉次顺序均分分配加工机器, 设  $n = \left\lfloor \frac{|\Omega|}{M_j} \right\rfloor, m = |\Omega| - nM_j, x_{ijk} = 1, (k-1)n + r_{k-1} + 1 \leq i \leq kn + r_k,$  其中当  $k \leq m$  时,  $r_k = k;$  当  $k > m$  时,  $r_k = m, 1 \leq k \leq M_j, j = 1, 2;$

**步骤 6** 将确定好的整数变量  $x_{ijk}$  和  $y_{irj}$  代入炼钢-连铸生产调度模型, 利用单纯形法或内点法求解, 可得到各个炉次在各个阶段相应的开始加工时间  $t_{ij},$  结束.

#### 4.5 构造相对最优可行解

松弛函数值在拉格朗日松弛水平算法迭代后期是小幅振荡下降的(由下面 4.6 小节中的定理 2 和定理 3 可知), 振荡幅度随着迭代次数增加越来越小, 达到终止条件时结束迭代. 松弛问题函数值的小幅振荡下降, 必然也导致基于其解的原问题函数值也是振荡或循环跳跃的. 为了得到此情况下的较优的原问题可行解, 需基于松弛问题最终迭代解, 再进行若干次迭代, 以得到相对最优可行解. 具体步骤如下:

**步骤 1** 由拉格朗日松弛水平算法, 可知最终的拉格朗日乘子  $\mu_{m+1}, F_{\text{rec}}^m$  和  $\delta_{l+1}.$  同时基于  $\mu_{m+1}$  得到松弛问题的解, 利用 4.4 小节构造可行解的方法得到原问题可行解和函数值  $f_0.$  设  $\bar{N} > 0 (\bar{N}$  为偶数),  $\bar{\mu}_0 \leftarrow \mu_{m+1}, \bar{\delta}_0 \leftarrow \delta_{l+1}, n \leftarrow 0;$

**步骤 2**  $F_{\text{lev}}^m \leftarrow F_{\text{rec}}^{m(l)} - \bar{\delta}_0, z_1 \leftarrow \bar{\mu}_0 - d_0, \bar{\mu}_1 \leftarrow P_{R^+}(z_1), d_0 \leftarrow t(F(\bar{\mu}_0) - F_{\text{lev}}^m) g_m / \|g_m\|^2;$

**步骤 3** 基于  $\bar{\mu}_1$  得到松弛的问题的解, 构造原问题的可行解和函数值  $f_1,$  如果  $f_1 < f_0,$  则停止; 否则,  $\bar{\delta}_0 \leftarrow \beta \bar{\delta}_0, \bar{\mu}_0 \leftarrow \bar{\mu}_1, n \leftarrow n + 1,$  转到步骤 4;<sup>1</sup>

**步骤 4** 如果  $n \leq \bar{N},$  则转到步骤 2; 否则, 停止.

#### 4.6 算法收敛性分析

**定理 1** 如果对于给定的  $m_0 \geq 1, \forall m \geq m_0, F_{\text{lev}}^m \geq \hat{F} > F^*,$  则拉格朗日松弛水平算法收敛到非最优解  $\bar{F}^*,$  即  $\lim_{m \rightarrow +\infty} F(\mu_m) = \lim_{m \rightarrow +\infty} F_{\text{lev}}^m = \bar{F}^*$  且  $\bar{F}^* \geq \hat{F} > F^*.$

**定理 2** 如果拉格朗日松弛水平算法无步骤 3 和步骤 4, 且对于给定的  $m_0 \geq 1, \forall m \geq m_0, F_{\text{lev}}^m < F^*,$  则其不能收敛, 即  $\sum_{m=1}^{+\infty} \|d_m\| = +\infty.$

定理 1 和定理 2 的证明见附录.

**定理 3** 如果  $\sum_{l=1}^{+\infty} h(l) = +\infty,$  则  $\lim_{k \rightarrow \infty} F_{\text{rec}}^k = \inf_{\mu \geq 0} F(\mu) = \inf_{\mu \geq 0} \max_{(x,y,t) \in X} G_L, X$  为  $G_L$  的可行域.<sup>2</sup>

**证明** 因为次梯度  $g_F(\mu_m) = (v_{irjk})_{N_\mu \times 1}$  有界, 因此利用文献[18,19]中次梯度的收敛性证明的方法, 以及定理 2 和定理 3 的结论, 即可得证. 证毕.

## 5 数值实例

### 5.1 基本数据

为了分析拉格朗日松弛水平算法的性能, 并且与文献[2-4]中所用的传统拉格朗日松弛算法进行比较. 由于本文与其问题模型不同, 且构造可行解方法也不一样, 所以对比测试时只是采用了其拉格朗日算法框

<sup>1</sup>在实际计算中,  $\beta \in (0, 1)$  可根据实际数据结果进行调整, 另外在步骤 4 中一般设  $h(l) = 1/(l+1).$  算法中  $R$  不能取太小, 否则会使  $\delta_l$  递减过快, 造成  $F_{\text{lev}}^m > F^*.$  由定理 2 可知, 这样会使得算法收敛到松弛函数的非最优解.

<sup>2</sup>由定理 2 可知,  $\delta_l$  不能取值太小, 否则算法可能会收敛到非最优解. 当  $F_{\text{lev}}^m < F^*$  时, 由定理 3 可知, 算法不能收敛, 而将在迭代后期震荡或循环跳跃. 因此, 需要减小  $\delta_l,$  以使  $F_{\text{lev}}^m$  逐渐逼近最优解. 算法中的微振荡检测这一步, 可以检测函数值是否循环跳跃, 以决定是否减小  $\delta_l.$  微振荡检测步骤中的  $W$  一般取 10 以内的整数, 本文取  $W = 4.$



架,其它方法都是基于本文的方法.

这两个算法都是在 Dell Precision T5500(CPU 主频为 2.93 GHz)机器上和 MATLAB 7.10 环境下编程实现.算法衡量的主要指标为对偶松弛函数值、原函数值和对偶间隙和运行时间.实验的相关参数和数据为(每类数据均随机生成十组数据进行比较).

为了便于统一比较结果,假定阶段数为 3,每个浇次中的炉次数相同,连浇阶段每个机器的浇次数相同,每个浇次中的炉次数为 8 个或 16 个;每个连铸机的上的浇次数为 1 个或 2 个;每个阶段的机器数取值然后,按照拉格朗日水平松弛算法步骤 1 到步骤 8(3.3 节),进行重复迭代,直到得到最终的对偶松弛函数值.经过计算可得到下界为  $q_2 = L^* = 2.7898 \times 10^5$ ;区间为  $[3, 5]$ ,且每个阶段机器数相同;驻留时间惩罚系数  $C_1 = 130$ ,提前惩罚系数  $C_2 = 100$ ,拖期惩罚系数  $C_3 = 10$ ;加工作业时间  $P_{i,1}$  是在区间  $[36, 40]$  均匀分布生成的整数,  $P_{i,2}$  是在区间  $[36, 50]$  均匀分布生成的整数,  $P_{i,3}$  是在区间  $[36, 48]$  均匀分布生成的整数;标准的运输时间  $T_{ij}$  是在区间  $[3, 6]$  均匀分布生成的整数,批次中相邻浇次的间隔时间  $T_B = 80$ ;计划开浇时间为

$$d_{b(k-1)+m+1} = d_{b(k-1)+m} + \sum_{i=n_{k,1}}^{n_{k,m}} P_{i,3} + T_B, d_{i_0} = \sum_{j=1}^2 P_{i_0,j} + \sum_{j=1}^2 T_{j,j+1},$$

其中  $i_0 = s_{b_{k-1}} + 1, \bar{k}_m = b(k-1) + m, n_{k,1} = s(\bar{k}_m), n_{k,m} = s(\bar{k}_m) - s(\bar{k}_m - 1), 1 \leq m < b(k) - b(k-1), 1 \leq k \leq M_3$ .

## 5.2 算法参数设置及其性能评价定义

为了便于计算,拉格朗日水平松弛算法(简称拉氏水平算法)中相关参数的取值为  $h(l) = 1/(l+1), \varepsilon_1 = \varepsilon_2 = \varepsilon_3 = 10^{-5}, R = 400 + 100 \times N, \mu_0 = \mathbf{0}, \mu_1 = 10^n \mathbf{g}_F(\mu_0) / \|\mathbf{g}_F(\mu_0)\|^2, \delta_1 = (F(\mu_0) + G(\mu_0))/C$ ,其中  $\mu_0$  为初始拉格朗日乘子,  $F(\mu_0)$  为松弛问题函数值,  $G(\mu_0)$  为基于  $\mu_0$  得到的可行解,  $C = 50 + 10N$  为一常数,  $N$  为浇次总数,  $n = \lceil \lg(\|\mathbf{g}_F(\mu_0)\|^2/U) \rceil, U = 1 \times 10^4$ .

算法测试对比时,两个算法的初始乘子均设为  $\mu_0 = \mathbf{0}$ .由于传统拉格朗日松弛算法(简称传统拉氏算法)不能保证收敛,因而都是以最大迭代次数来终止迭代,本文中其终止迭代次数为 500.两个拉格朗日算法除了迭代方式不一样以外,其它均相同,即连续变量子问题求解、整数变量子问题求解与构造可行解的方法一样(传统拉氏算法无构造最优可行解这一步).

记传统拉氏算法的松弛函数最终迭代值为  $\bar{q}_{\text{end}}$ (简记为  $q_{11}$ ),其对应的可行原函数值记为  $\bar{f}_{\text{end}}$ (简记为  $f_{11}$ ),迭代中的最优松弛函数值记为  $\bar{q}_{\text{best}}$ (简记为  $q_{12}$ ),其对应的可行原函数值为  $\bar{f}_{\text{best}}$ (简记为  $q_{12}$ );拉氏水平算法松弛函数最终迭代值  $f_{\text{end}}$ (简记为  $f_2$ ),其对应的可行原函数值  $q_{\text{end}}$ (简记为  $q_2$ ).同时记传统拉氏算法的运行时间为  $t_1$ ,拉氏水平算法运行时间为  $t_2$ .本文定义拉格朗日算法的相关对偶间隙 gap 值为

$g_{11} = (\bar{f}_{\text{end}} - \bar{q}_{\text{end}})/\bar{q}_{\text{end}}, g_{12} = (\bar{f}_{\text{best}} - \bar{q}_{\text{end}})/\bar{q}_{\text{end}}, g_{13} = (\bar{f}_{\text{best}} - \bar{q}_{\text{best}})/\bar{q}_{\text{best}}, g_2 = (f_{\text{end}} - q_{\text{end}})/q_{\text{end}}$ .这里  $g_{11}$  表示传统拉氏算法的最终迭代的对偶间隙值,  $g_{12}$  表示传统拉氏算法中的对偶函数最优值与最终迭代的可行原函数值的对偶间隙(文献[1-3]的对偶间隙为此定义),  $g_{13}$  表示传统拉氏算法在得到最优可行原函数解时所对应的对偶间隙.由于上述三个对偶间隙往往不相同,因而从这三个对偶间隙值可以全面地衡量传统拉氏算法的计算效果.

## 5.3 计算结果

### 5.3.1 简单实例

表 1 给出了 3 个浇次和 24 个炉次的加工时间( $P_{i,j}$ )数据,其中每个浇次中含有 8 个炉次.浇次 1 中的炉次序号为 1-8,浇次 2 中的炉次序号为 9-16,浇次 3 中的炉次序号为 17-24.炉次在各阶段间的传输时间为  $T_{1,2} = 4, T_{2,3} = 5$ .

拉格朗日松弛水平算法具体求解过程如下:首先,求解对偶函数  $L(\mu_1)$ (定义见 4.1 节),此时

$$G_L(\mu_1) = 130 \sum_{i=1}^{24} (t_{i,3} - t_{i,1}) + 100 \sum_{n=1}^3 t_{s(n-1)+1,3}^1 + 10 \sum_{n=1}^3 t_{s(n-1)+1,3}^u + F_3,$$

其中  $F_3 = -\sum_{j=1}^2 \sum_{k=1}^3 \sum_{i=1}^{24} \sum_{r=1, r \neq i}^{24} \mu_{irjk}(t_{rj} - t_{ij} - P_{ij} + (3 - x_{ijk} - x_{rjk} - y_{irj})U)$ ,  $\mu_1 = 10^8 \mathbf{g}_F(0) / \|\mathbf{g}_F(0)\|^2$ , 次梯度  $\mathbf{g}_F(\boldsymbol{\mu})$  定义见 4.3 节. 其它相关参数为  $C = 80, R = 700, h(1) = 1/2, F(\boldsymbol{\mu}_1) = -L(\boldsymbol{\mu}_1) = 6.0781 \times 10^7, G(\boldsymbol{\mu}_1) = 2.9164 \times 10^5, \delta_1 = 7.6341 \times 10^4$ .

然后, 按照拉格朗日水平松弛算法步骤 1 到步骤 8(3.3 节), 进行重复迭代, 直到得到最终的对偶松弛函数值. 经过计算可得到下界为  $q_2 = L^* = 2.7898 \times 10^5$ ;

表 1 实验数据  
Table 1 Data for testing

炉次序号	加工时间/min			炉次序号	加工时间/min		
	第一阶段	第二阶段	第三阶段		第一阶段	第二阶段	第三阶段
1	37	47	43	13	39	46	44
2	37	36	43	14	38	39	44
3	38	37	46	15	37	41	40
4	36	38	46	16	38	45	41
5	40	45	44	17	36	47	43
6	40	46	42	18	37	37	40
7	38	45	46	19	40	49	46
8	38	42	43	20	40	47	40
9	37	44	42	21	38	43	40
10	40	40	47	22	36	42	40
11	37	46	47	23	37	42	40
12	36	39	44	24	37	40	42

最后, 根据最终迭代的所得的拉格朗日乘子, 由步骤 9(4.3 节)可得到相对最优可行解为  $f_2 = G^* = 2.8798 \times 10^5$ , 其对偶间隙为  $g_2 = (f_2 - q_2)/q_2 \times 100\% = 3.23\%$ .

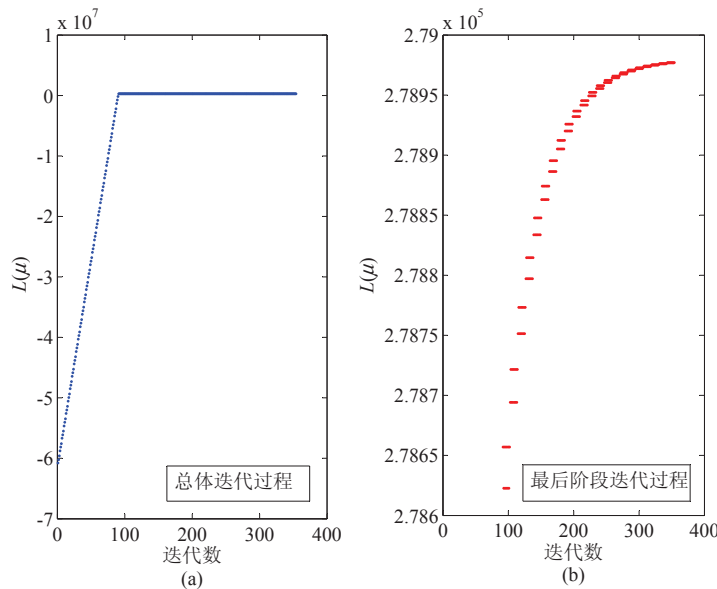


图 2 拉格朗日松弛水平算法迭代过程(K 为迭代次数)

Fig. 2 The iteration process of Lagrangian relaxation level algorithm(K is number of iterations)

图 2 给出了拉格朗日松弛水平算法的对偶函数值  $L(\boldsymbol{\mu})$ (下界)迭代过程, 其中图 2(a)描述的是算法总体迭代过程, 由于函数值在迭代后期较为接近, 因而用图 2(b)描述了算法最后阶段迭代过程.

由图 2(a)可知, 拉格朗日松弛水平算法继承了传统拉格朗日算法的优点, 在开始阶段上升很快, 但接近最优点时, 速度则较慢. 由图 2(b)可知, 算法在最后迭代阶段, 不断调整估计的函数值, 而且调整幅度越来越小, 最终收敛于最优对偶函数值. 将最终得到的下界与可行解进行比较, 可知此算法得到的解离真正的最优解的距离不会超过  $G^* - L^* = 9000$ , 即间隙小于 3.23%.

5.3.2 多组不同实例

为了进一步测试算法的性能,计算了其它不同类型的数据.在数值实例比较中,每种问题都随机生成十组测试数据进行计算比较,记录每组数据的最好值、最坏值和平均值,具体计算结果见表 2、表 3 和表 4.

表 2 传统拉格朗日松弛算法的计算结果

Table 2 The result of traditional Lagrangian relaxation algorithm

炉次数	浇次数	机器数	$q_{11}$			$f_{11}$			$f_{12}$		
			最佳值	最差值	平均值	最佳值	最差值	平均值	最佳值	最差值	平均值
24	3	3	283 138	<b>273 489.4</b>	<b>279 371.9</b>	281 400	292 250	287 779	281 290	290 040	287 399
32	4	4	382 573	372 287	376 256	386 880	398 200	391 807	386 470	395 420	390 146
40	5	5	474 935	462 799	467 896	482 210	497 740	492 234	480 680	495 090	489 093
48	6	3	570 795	555 626	562 100	596 760	615 610	603 191	595 060	614 080	601 484
64	8	4	757 378	743 138	750 220	817 230	833 600	824 568	816 410	831 370	823 599
80	10	5	897 611	837 063	878 171	1 050 600	1 078 940	1 063 033	1 048 880	1 066 230	1 057 223
48	3	3	564 849	555 819	561 613	574 360	584 810	580 945	574 300	583 080	580 306
64	4	4	755 617	743 496	750 403	776 950	794 920	785 582	777 640	789 810	784 836
80	5	5	875 516	842 392	857 878	990 320	1 011 320	999 132	982 120	1 003 360	995 153
96	6	3	1 042 445	1 013 146	1 029 607	1 203 640	1 215 770	1 208 172	1 203 140	1 213 160	1 206 657
128	8	4	1 424 090	1 336 698	1 383 622	1 646 050	1 678 570	1 661 395	1 643 760	1 671 880	1 657 925
160	10	5	1 804 687	1 756 544	1 778 483	2 124 170	2 158 770	2 137 922	2 116 700	2 140 900	2 131 110

注:加粗的数字表示所比较算法中的最好计算结果.

表 3 传统拉格朗日松弛算法的对偶间隙与运行时间

Table 3 The dual gaps and running times of the traditional Lagrangian relaxation algorithm

炉次数	浇次数	机器数	$g_{11}$			$g_{12}$			$g_{13}$			$t_1(s)$		
			最佳值	最差值	平均值	最佳值	最差值	平均值	最佳值	最差值	平均值	最佳值	最差值	平均值
24	3	3	0.027 8	0.038 4	0.030 1	<b>0.024 4</b>	<b>0.034 0</b>	<b>0.028 8</b>	0.024 7	0.035 7	0.029 5	<b>2.380 0</b>	<b>3.29</b>	<b>2.81</b>
32	4	4	0.034 7	0.054 0	0.041 3	<b>0.033 6</b>	<b>0.039 9</b>	<b>0.037 0</b>	0.049 8	0.120 0	0.070 1	23.080 0	24.68	24.01
40	5	5	0.038 4	0.069 8	0.052 0	<b>0.038 2</b>	<b>0.055 2</b>	<b>0.045 3</b>	0.042 0	0.125 4	0.061 6	<b>14.070 0</b>	30.65	28.18
48	6	3	0.069 2	0.078 5	0.073 1	<b>0.065 2</b>	<b>0.075 9</b>	<b>0.070 1</b>	0.066 6	0.095 9	0.076 1	<b>11.300 0</b>	37.07	29.68
64	8	4	0.097 0	0.101 3	0.099 1	<b>0.096 3</b>	<b>0.099 5</b>	<b>0.097 9</b>	0.096 8	0.121 9	0.101 8	<b>13.720 0</b>	48.87	<b>20.41</b>
80	10	5	0.171 3	0.282 2	0.211 1	0.169 0	0.263 7	0.205 0	0.185 6	4.980 2	1.252 7	67.030 0	70.03	68.05
48	3	3	0.032 1	0.038 4	0.034 4	<b>0.031 5</b>	<b>0.035 4</b>	<b>0.033 3</b>	0.032 1	0.308 2	0.086 6	<b>15.680 0</b>	31.63	22.88
64	4	4	0.045 0	0.052 0	0.046 9	<b>0.043 6</b>	<b>0.048 7</b>	<b>0.045 9</b>	0.045 9	0.063 5	0.049 6	<b>21.090 0</b>	44.49	36.12
80	5	5	0.149 8	0.188 0	0.164 8	0.145 4	0.188 6	0.160 7	0.155 5	0.461 8	0.260 7	62.420 0	63.98	63.05
96	6	3	0.154 6	0.189 7	0.173 5	0.155 4	0.190 5	0.172 8	0.155 4	8.535 6	1.886 7	74.290 0	79.48	76.77
128	8	4	0.176 0	0.234 1	0.201 2	0.172 9	0.235 0	0.199 5	0.191 1	1.729 0	0.447 5	125.810 0	131.48	128.29
160	10	5	0.189 2	0.226 4	0.202 2	0.185 8	0.216 9	0.198 9	0.210 4	4.366 0	0.696 9	197.000 0	202.25	199.35

注:加粗的数字表示所比较算法中的最好计算结果.

5.4 计算结果分析

从数值实例计算结果可知,本文提出的拉格朗日松弛水平算法(简称拉氏水平算法)从计算时间和松弛函数的最优值优于传统的拉格朗日算法(简称传统拉氏算法),对偶间隙整体上优于传统的拉格朗日松弛算法.具体分析如下:

1) 拉氏水平算法的对偶函数值优于传统拉氏算法.从表 2 和表 3 的 12 组实例结果的统计结果可知,拉氏水平算法的对偶函数值均大于传统拉氏算法,即得到了更好的下界.尤其问题规模越大时,两个算法的值差距越大;

2) 问题规模越大,拉氏水平算法的对偶间隙优于传统拉氏算法.从表 3 和表 4 的 12 组实例结果可知,在问题规模较小时,虽然传统拉氏算法的对偶间隙小于拉氏水平算法,但两者差别一般不大,即  $g_2 - g_{12} \leq 0.01$ ;但随着问题规模增大,两者相差甚至达到  $\max(g_{12} - g_2) = 0.115 7$ (炉次数为 128 时).这说明问题规模越大,拉氏水平算法的所得到的间隙相对更优;

3) 传统拉氏算法的对偶间隙计算结果不稳定,而拉氏水平算法的对偶间隙计算结果较为稳定.比较表 2 中的  $g_{11}$ 、 $g_{12}$  和  $g_{13}$  统计结果,可以发现,三者基本不相同,尤其  $g_{13}$  的结果波动最大.主要原因在于传统拉氏算法并不能保证收敛;

4) 问题规模越大, 拉氏水平算法的求解时间的优势越明显. 从表3和表4中可以看出, 在炉次数为160时, 传统拉氏算法比拉氏水平算法多38s, 相差达25%. 另外, 从表3中的运行时间可以看出, 对于同一组数据, 拉氏水平算法的计算时间比较平稳, 而传统拉氏算法运行时间波动较大.

表4 拉格朗日松弛水平算法的计算结果

Table 4 The computational results of the Lagrangian relaxation level algorithm

炉次数	浇次数	机器数	$q_2$			$f_2$			$g_2$			$t_2(s)$		
			最佳值	最差值	平均值	最佳值	最差值	平均值	最佳值	最差值	平均值	最佳值	最差值	平均值
24	3	3	273 489	<b>283 139</b>	279 372	284 760	295 270	290 405	0.028 5	0.046 6	0.039 5	7.49	8.15	7.83
32	4	4	<b>373 019</b>	<b>382 697</b>	<b>376 595</b>	390 350	401 980	396 452	0.045 7	0.057 9	0.052 7	<b>12.57</b>	<b>13.96</b>	<b>13.23</b>
40	5	5	<b>462 794</b>	<b>475 018</b>	<b>467 934</b>	491 950	506 640	497 709	0.056 0	0.071 1	0.063 6	16.27	<b>17.14</b>	<b>16.53</b>
48	6	3	<b>555 628</b>	<b>570 795</b>	<b>562 099</b>	598 020	615 460	607 542	0.076 3	0.086 5	0.080 9	21.60	<b>24.93</b>	<b>22.54</b>
64	8	4	<b>743 134</b>	<b>757 374</b>	<b>750 217</b>	826 570	845 810	836 462	0.110 5	0.120 2	0.115 0	31.30	<b>34.8</b>	32.58
80	10	5	<b>931 707</b>	<b>942 167</b>	<b>936 647</b>	1 064 750	1 084 060	1 075 564	<b>0.142 8</b>	<b>0.155 6</b>	<b>0.148 3</b>	<b>54.23</b>	<b>57.91</b>	<b>55.54</b>
48	3	3	<b>555 818</b>	<b>564 848</b>	<b>561 651</b>	576 880	586 730	583 213	0.035 5	0.041 8	0.038 4	18.55	<b>19.15</b>	<b>18.88</b>
64	4	4	<b>743 498</b>	<b>755 614</b>	<b>750 412</b>	783 200	799 220	792 449	0.053 4	0.057 7	0.056 0	26.29	<b>27.43</b>	<b>26.79</b>
80	5	5	<b>930 510</b>	<b>944 396</b>	<b>939 034</b>	995 480	1 012 690	1 005 907	<b>0.068 2</b>	<b>0.073 9</b>	<b>0.071 2</b>	<b>46.75</b>	<b>49.11</b>	<b>47.86</b>
96	6	3	<b>1 120 333</b>	<b>1 129 932</b>	<b>1 123 636</b>	1 205 490	1 218 990	1 215 336	<b>0.075 2</b>	<b>0.084 4</b>	<b>0.081 6</b>	<b>65.28</b>	<b>70.4</b>	<b>67.65</b>
128	8	4	<b>1 486 413</b>	<b>1 509 580</b>	<b>1 498 081</b>	1 649 380	1 685 000	1 669 247	<b>0.108 6</b>	<b>0.119 3</b>	<b>0.114 3</b>	<b>104.9</b>	<b>113.94</b>	<b>110.09</b>
160	10	5	<b>1 858 771.3</b>	<b>1 882 513</b>	<b>1 872 024</b>	2 137 630	2 164 350	2 150 479	<b>0.142 6</b>	<b>0.155 9</b>	<b>0.148 8</b>	<b>157.8</b>	<b>165.53</b>	<b>161.37</b>

注: 加粗的数字表示所比较算法中的最好计算结果.

## 6 结束语

本文建立了炼钢-连铸生产调度的0-1混合整数规划模型. 求解时, 将目标函数中的非线性的目标函数转化成线性的目标函数. 通过松弛资源析取约束, 解除连续变量和整数变量之间的耦合关系, 将松弛问题分解成两个容易求解的子问题. 提出了一种拉格朗日松弛水平算法, 采用对偶松弛函数下降水平控制方法, 以保证其收敛于对偶松弛问题最优解, 并给出了收敛性证明.

## 参考文献:

- [1] Tang L, Luh P B, Liu J, et al. Steelmaking process scheduling using lagrangian relaxation[J]. International Journal of Production Research, 2002, 40(1): 55-70.
- [2] Tang L, Xuan H, Liu J. A new Lagrangian relaxation algorithm for hybrid flowshop scheduling to minimize total weighted completion time[J]. Computers and Operations Research, 2006, 33(11): 3344-3359.
- [3] Xuan H, Tang L. Scheduling a hybrid flow-shop with batch production at the last stage[J]. Computers and Operations Research, 2007, 34(9): 2718-2733.
- [4] Pacciarelli D, Pranzo M. Production scheduling in a steelmaking-continuous casting plant[J]. Computers and Chemical Engineering, 2004, 28(12): 2823-2835.
- [5] 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.
- [6] 刘光航, 李铁克. 炼钢-连铸生产调度模型及启发式算法[J]. 系统工程学报, 2002, 20(6): 44-48.  
Liu Guanghang, Li Tiek. A steelmaking-continuous casting production scheduling model and its heuristic algorithm[J]. Journal of Systems Engineering, 2002, 20(6): 44-48. (in Chinese)
- [7] 李铁克, 周健, 孙林. 连铸连轧和冷装热轧并存环境下的炼钢-连铸生产调度模型与算法[J]. 系统工程理论与实践, 2006, 26(6): 117-123.  
Li Tiek, Zhou Jian, Sun Lin. Model and algorithm for steelmaking-continuous casting scheduling in DHCR and CCR Environment[J]. Systems Engineering: Theory & Practice, 2006, 26(6): 117-123. (in Chinese)
- [8] 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, 17(5): 68-74.  
Li Tiek, Su Zhixiong. Two-stage genetic algorithm for SM-CC production scheduling[J]. Chinese Journal of Management Science, 2009, 17(5): 68-74. (in Chinese)
- [9] 庞新富, 俞胜平, 罗小川, 等. 混合Jobshop炼钢-连铸重调度方法及其应用[J]. 系统工程理论与实践, 2012, 32(4):826-838.  
Pang Xinfu, Yu Shengping, Luo Xiaochuan, et al. Hybrid job shop rescheduling method and its application for steelmaking-casting. Systems Engineering: Theory & Practice[J]. 2012, 32(4): 826-838. (in Chinese)

- [10] 俞胜平, 柴天佑, 郑秉霖. 炼钢-连铸混合智能优化调度方法及应用[J]. 系统工程学报, 2010, 25(3): 379-386.  
Yu Shengping, Chai Tianyou, Zheng Binglin. Hybrid intelligent optimal scheduling method for steelmaking and continuous casting and its application[J]. Journal of Systems Engineering, 2010, 25(3): 379-386. (in Chinese)
- [11] Pacciarelli D. The alternative graph formulation for solving complex factory scheduling problems[J]. International Journal of Production Research, 2002, 40(15): 3641-3653.
- [12] Wang W, Luh P B, Yan J H, et al. An improved Lagrangian relaxation method for discrete optimization applications[C] // IEEE International Conference on Automation Science and Engineering, Washington, IEEE, 2008: 359-364.
- [13] Zhao X, Luh P B, Wang J. Surrogate gradient algorithm for Lagrangian relaxation[J]. Journal of Optimization Theory and Applications, 1999, 100(3): 699-712.
- [14] Sun T, Zhao Q, Luh P B. On the surrogate gradient algorithm for Lagrangian relaxation[J]. Journal of Optimization Theory and Applications, 2007, 133(3): 413-416.
- [15] Guan X, Zhai Q, Lai F. New Lagrangian relaxation based algorithm for resource scheduling with homogeneous sub-problems[J]. Journal of Optimization Theory and Applications, 2002, 113(1): 65-82.
- [16] Tang L, Liu J, Rong A, et al. A mathematical programming model for scheduling steelmaking continuous casting production[J]. European Journal of Operational Research, 2000, 120(2): 423-435.
- [17] Bertsekas D P. Nonlinear Programming[M]. Massachusetts: Athena Scientific, 1999.
- [18] Nedic A, Bertsekas D P. Incremental subgradient methods for non-differentiable optimization[J]. SIAM Journal on Optimization, 2001, 12(1): 109-138.
- [19] Goffin J L, Kiwiel K C. Convergence of a simple subgradient level method[J]. Mathematical Programming, 1999, 85(1): 207-211.

### 作者简介:

毛 坤(1984—), 男, 湖北黄冈人, 博士生, 研究方向: 钢铁计划调度与优化, Email: mao\_kun@126.com;

潘全科(1971—), 男, 山东聊城人, 博士, 教授, 博士生导师, 研究方向: 复杂生产过程的优化调度理论与方法等, Email: panquanke@mail.neu.edu.cn;

庞新富(1978—), 男, 辽宁营口人, 博士生, 研究方向: 钢铁生产调度及其应用软件开发, Email: pangxinfu@163.com.

### 附 录

为了证明定理1先给出引理1.

引理1 令  $\Psi = \{\mu | \mu \geq 0, \mu \in R^K\}$ , 其中  $K = |\Omega| (|\Omega| - 1) \sum_{j=1}^2 M_j$ , 令  $s_m = t(F(\mu_m) - F_{lev}^m) / \|\mathbf{g}_m\|^2$ , 则  $\forall \mu \in \Psi$ , 有

$$\|\mu_{m+1} - \mu\|^2 \leq \|\mu_m - \mu\|^2 - 2s_m(F(\mu_m) - F(\mu)) + s_m^2 \|\mathbf{g}_m\|^2.$$

证明 因为  $F(\mu)$  为凸函数, 所以  $F(\mu) - F(\mu_m) \geq \mathbf{g}_m^T(\mu - \mu_m)$ , 又  $\|\mu_{m+1} - \mu\|^2 = |P_{R^+}(\mu_m - s_m \mathbf{g}_m) - P_{R^+}(\mu)|^2 \leq q \|\mu_m - \mu\|^2 - 2s_m \mathbf{g}_m^T(\mu_m - \mu) + s_m^2 \|\mathbf{g}_m\|^2$ , 因此由上面两式可知

$$\|\mu_{m+1} - \mu\|^2 \leq \|\mu_m - \mu\|^2 - 2s_m(F(\mu_m) - F(\mu)) + s_m^2 \|\mathbf{g}_m\|^2. \quad \text{证毕.}$$

定理1的证明

证明 因为  $F(\mu)$  是连续函数, 因此必存在一个  $\tilde{\mu} \in \Psi$ , 使得  $F_{lev}^m \geq \hat{F} > F(\tilde{\mu}) > F^*$ . 由引理1可知

$$\begin{aligned} \|\mu_{m+1} - \tilde{\mu}\|^2 &\leq \|\mu_m - \tilde{\mu}\|^2 - 2s_m(F(\mu_m) - F(\tilde{\mu})) + s_m^2 \|\mathbf{g}_m\|^2 \\ &\leq \|\mu_m - \tilde{\mu}\|^2 - 2s_m(F(\mu_m) - F_{lev}^m) + s_m^2 \|\mathbf{g}_m\|^2 \\ &\leq \|\mu_m - \tilde{\mu}\|^2 - t(2-t) \frac{(F(\mu_m) - F_{lev}^m)^2}{\|\mathbf{g}_m\|^2}. \end{aligned}$$

因为  $\mathbf{g}_m$  有界, 即  $\|\mathbf{g}_m\| < M$ ,  $M$  为足够大常数, 则有

$$\frac{t(2-t)}{M^2} \sum_{m=0}^{+\infty} (F(\mu_m) - F_{lev}^m)^2 \leq \|\mu_0 - \tilde{\mu}\|^2.$$

显然  $\lim_{m \rightarrow +\infty} F(\mu_m) = \lim_{m \rightarrow +\infty} F_{lev}^m = \bar{F}^*$ , 且  $\bar{F}^* \geq \hat{F} > F^*$ . 证毕.

定理2的证明.

证明 因为  $\mathbf{g}_m$  有界, 即  $\|\mathbf{g}_m\| < M$ ,  $M$  为足够大常数, 对任意大的  $N > 0$ , 有

$$\sum_{m=0}^N \|\mathbf{d}_m\| = \sum_{m=0}^N \frac{F(\mu_m) - F_{lev}^m}{\|\mathbf{g}_m\|} \geq \frac{1}{M} \sum_{m=0}^N (F(\mu_m) - F_{lev}^m) \geq \frac{N+1}{M} (F^* - F_{lev}^m).$$

因为  $F_{lev}^m < F^*$ , 所以当  $N \rightarrow +\infty$  时,  $\sum_{m=0}^N \|\mathbf{d}_m\| \rightarrow +\infty$ . 证毕.