

考虑板坯设计的组炉优化模型

杜斌^{1,2}, 朱俊¹, 贾树晋², 刘士新¹

(1. 东北大学 信息科学与工程学院, 辽宁 沈阳 110004; 2. 宝钢研究院自动化所, 上海 201900)

摘要: 针对炼钢生产中的组炉优化问题建立了一种考虑板坯设计的混合整数规划模型, 并提出了一种基于非二分图匹配算法、二分图匹配算法、装箱算法、网络最大流算法的启发式求解算法。该算法首先使用非二分图匹配算法确定炉次, 然后使用二分图匹配算法和装箱算法将剩余合同匹配到已有炉次中, 最后使用网络最大流算法调整炉次中合同对应的板坯质量。试验结果表明利用该算法可以在较短的时间内给出较优的组炉方案, 为计划员提供决策支持。

关键词: 炼钢; 组炉优化; 板坯设计; 启发式

中图分类号: O223

文献标志码: A

A Charge Optimization Model with a Consideration of Slab Design

DU Bin^{1,2}, ZHU Jun¹, JIA Shujin², LIU Shixin¹

(1. School of Information Science and Engineering, Northeastern University, Shenyang 110004, China; 2. Department of Automation, R & D Institute of Baosteel, Shanghai 201900, China)

Abstract: As for the charge optimization problem in steel-making process, a mixed integer programming mathematical model, which takes into account of the slab design, is proposed. Moreover, a heuristic algorithm, which is based on nonbipartite matching, bipartite matching, bin-packing and network maximum flow algorithms, is also proposed to solve the above optimization model. First, a nonbipartite matching algorithm is used to determine the charges. Then, the bipartite matching and bin packing algorithms are used to match the orders to the existing charges. Finally, the network maximum flow algorithm is used to adjust the weight of slabs assigned in the charges. Experimental results indicate that the proposed algorithm can obtain an optimized matching solution within reasonable time, which can provide enough decision support for planners.

Key words: steel-making; charge optimization; slab design; heuristic

炼钢组炉问题属于炼钢-连铸段的生产批量计划中的炉次计划问题^[1-2]。由于工艺原因, 计划人员必须将不同的合同组织到同一炼钢炉进行生产, 炉中合同必须满足如下几个条件: ①炉中各合同所采用的钢级(steel grade)必须相同; ②炉中各合同所采用板坯规格(厚度、宽度)必须相同, 这是下游的连铸工序约束; ③设炉中某合同总质量为 X , 则必存在某整数 N , 使 X/N 落在合同要求的板坯质量上下限范围内; ④炉内合同总量不得超过炼钢炉最大生产容量, 在炼钢生产中, 若某炉的合同量不满一炉时, 仍然需要以一炉的量进行组织生产, 一炉中没有合同对应的多余产量称为余材^[3]。

组炉优化的一般流程如下: 首先按照合同要求进行板坯规格设计, 确定合同所需要的板坯的数量以及板坯规格, 之后按炼钢组炉的要求将不同的板坯归并到一个炼钢炉进行生产, 并确定板坯所选择的钢级(即对应炼钢炉的钢级)。各合同均设计有一系列可选择的钢级, 分别称为主钢级、副钢级, 使用主钢级的生产成本最低, 但有时候为减少余材的产生, 也会选择副钢级进行生产^[4]。

对于组炉优化问题, 学者们做了很多研究, 其中 Tang 等^[2]研究了炼钢组炉优化与连铸组浇优化问题, 该模型考虑了多个目标(数量级不同)并通过加权方法转化为单目标, 但其权重系数的确定是一大难题。黄可为等^[3]以板坯为基础分别建立了不考虑副钢级取代成本与考虑副钢级取代成本的数学模型, 给出了基于动态规划算法的求解方案。朱俊等^[4]则以合同为基础建立了多目标混合规划数学模型, 并给出了结合 PBIL (population-based incremental learning) 算法与网络最大流算法的求解方案, 该方法未考虑板坯质量限制。唐立新等^[5]同样从板坯出发, 给出了综合考虑交货期与钢级取代成本的对应

模型,并给出了基于遗传算法求解方案.本文从合同出发,同时考虑板坯设计^[6-7]的合理性,建立统一的多目标混合整数规划模型,并结合匹配算法、网络最大流算法以及装箱算法设计出一套启发式算法.试验分析表明,该算法能快速找到组炉优化问题的较优解,适合现场应用.

1 优化模型

炼钢组炉问题可描述为:有 M 个规格要求相同且属于同一钢级序列的合同 $O = \{o_1, o_2, \dots, o_i, \dots, o_M\}$, 钢级序列为 $G = \{g_1, g_2, \dots, g_k\}$, 合同 o_i 所允许的钢级集合为 $G_i, G_i \subseteq G$, 其中一个钢级为主钢级,其余为副钢级,订货量为 m_i , 最少供货量为 $m_{i,\min}$, 最大供货量为 $m_{i,\max}$, 合同要求板坯最小质量 $U_{i,\min}$, 最大质量为 $U_{i,\max}, i=1, 2, \dots, I$. 炼钢炉每炉最多炼钢质量为 M_{\max} , 最少炼钢质量为 M_{\min} , 若某炉炼钢质量在 $M_{\min} \sim M_{\max}$ 范围内,则认为余材量为零. 要求合理组织炉次,使炼钢余材量最少,并尽量减少副钢级的使用.

为方便建模,现假设至多需要 J 个空炉,各炉具有唯一的冶炼钢级属性,即若某合同归并到某炉进行冶炼则必须按照该炉的冶炼钢级进行生产. 设各合同采用不同炉次进行冶炼的单位代价成本为 $c_{ij}, i=1, 2, \dots, I, j=1, 2, \dots, J$, 若合同 i 不能采用炉 j 进行生产,则 $c_{ij} = +\infty$. 据此,建立以下混合整数规划数学模型:

$$\min \sum_i \sum_j x_{ij} c_{ij} \tag{1}$$

$$\min \sum_j (M_j - \sum_i x_{ij}) \tag{2}$$

s. t.

$$m_{i,\min} \leq \sum_j x_{ij} \leq m_{i,\max} \tag{3}$$

$$(\sum_i x_{ij}) T^{-1} \leq l_j \leq (\sum_i x_{ij} + T) T^{-1} \tag{4}$$

$$M_j T^{-1} \leq l_j \leq (M_j + T) T^{-1} \tag{5}$$

$$l_j \cdot M_{\min} \leq M_j \leq l_j \cdot M_{\max} \tag{6}$$

$$\sum_i x_{ij} \leq M_j \tag{7}$$

$$N_{ij} \cdot U_{i,\min} \leq x_{ij} \leq N_{ij} \cdot U_{i,\max} \tag{8}$$

$$N_{ij} \in \mathbf{Z} \tag{9}$$

$$l_j \in \{0, 1\} \tag{10}$$

式中: x_{ij} 为第 i 个合同匹配到第 j 个炉的钢的质量; M_j 为中间决策变量,表示第 j 炉的炼钢质量,当 l_j

$= 0$ 时 $M_j = 0$, 当 $l_j = 1$ 时 $M_{\max} \geq M_j \geq \max(\sum_i x_{ij}, M_{\min})$; T 为某极大常数; l_j 为变量,标志第 j 炉的使用情况,若有合同采用该炉进行冶炼,则为 1, 否则为 0; N_{ij} 为第 i 个合同在第 j 炉中的板坯数.

该模型共有 2 个目标,分别是:目标 1 副钢级成本最低,目标 2 炼钢总余材量最少.

各约束式含义如下:式(3)表示合同生产量在其供货公差上下限内;式(4)~(7)表示若某炉生产量大于零 ($\sum_i x_{ij} > 0$), 则 $l_j = 1, M_j \geq \max(\sum_i x_{ij}, M_{\min})$, 否则 $l_j = 0, M_j = 0$; 式(8)、式(9)表示板坯的单重约束;式(10)为 l_j 的变量约束.

上述模型是一个多目标优化模型,在实际生产中这 2 个目标的优先级不同,一般来说成本的优先级高于余材,主要是因为余材可以通过余材充当等方式进行消耗,而高的生产成本则难以通过其他方法挽回.

若不考虑钢级约束,则该问题可转化为使余材量最少的单目标优化问题,为 Bin-packing 问题的一种特殊形式. 由于 Bin-packing 问题属于 NP-hard 问题^[8-9], 故本问题也为 NP-hard 问题.

2 一种基于匹配的启发式算法

求解 NP-hard 问题的常用算法主要包括遗传算法、精确算法(小规模问题)以及启发式算法等. 由于启发式算法具有计算速度快、灵活性强、可借鉴现场经验等优点,因此从工业应用角度来讲应用最广. 基于此本文设计了一种基于最大匹配原理、装箱算法及网络最大流的启发式算法. 算法流程如图 1 所示,其中非二分图匹配算法用于确定炉次;二分图算法与装箱算法用于将剩余合同匹配到已有的炉次中;网络最大流算法用于调整炉次中合同对应的板坯质量,以减少余材量;最后过滤筛选掉余材量过大的炉次.

步骤 1:非二分图匹配. 对于给定的合同,可按照以下方法构造一个合同兼容性图 $G = (V, E)$: 对于每个合同 o_i , 存在一个节点 $v_i \in V$. 如果 2 个合同 o_i, o_j 可以组成一炉(属于同一钢级序列),则增加边 $e = (v_i, v_j)$, 合同 o_i, o_j 的匹配量可通过求解以下混合规划问题得到.

$$\max x_i + x_j$$

$$\text{s. t. } \begin{cases} n_i U_{i,\min} \leq x_i \leq n_i U_{i,\max} \\ n_j U_{j,\min} \leq x_j \leq n_j U_{j,\max} \\ x_i + x_j \leq M_{\max} \\ x_i \leq m_{i,\max} \\ x_j \leq m_{j,\max} \\ n_i, n_j \in \mathbf{Z} \end{cases}$$

其中: x_i, x_j 为合同 o_i, o_j 的质量; n_i, n_j 为合同 o_i, o_j 的板坯数. $e=(v_i, v_j)$ 的权重 w_{ij} 代表 2 个合同 o_i, o_j 的匹配度, 即 o_i, o_j 在一炉中冶炼的代价, 代价越小越好. 假设合同 o_i 的钢级优于 o_j , 那么 o_i, o_j 组成的炉次必须按照 o_i 的钢级生产, 反之, 则炉次须按照 o_j 的钢级生产, 生产中称之为“以优充次”. 设 c_{ij} 表示合同 o_i, o_j 组成一炉的单位代价, 则令 $c'_{ij} = \max_{i,j} \{c_{ij}\} - c_{ij}$, 若 o_i 的钢级优于 o_j , 那么 $w_{ij} = c'_{ij} x_j$, 否则 $w_{ij} = c'_{ij} x_i$.

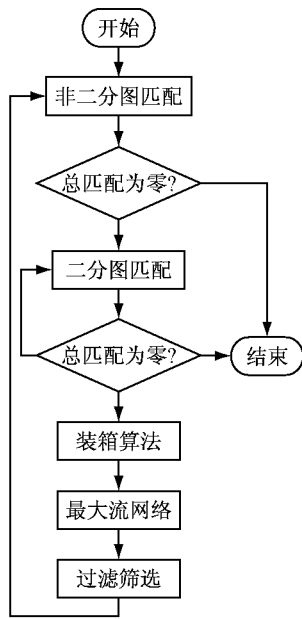


图 1 基于匹配的启发式算法
Fig. 1 A matching-based heuristic

如图 2 所示, 非二分图 G 的最大权重匹配 M 意味着: ① 匹配的合同 o_i, o_j 是兼容的, 即可以组成一炉; ② 每个合同最多只能出现在一炉中; ③ 匹配的总权重最大. 通过非二分图匹配算法^[7] 可以获得 G 的最大权重匹配 M , 得到 $|M|$ 个初始炉次. 如果总匹配等于零, 结束程序, 否则, 转步骤 2.

步骤 2: 二分图匹配. 该步骤循环地构建一个以剩余合同与炉次为节点的二分图并寻找其最大权重二分图匹配^[10], 算法步骤如下: ① 根据步骤 1 得到的结果, 令 R_i 表示合同 o_i 的剩余质量, R_j 表示炉次 j 的剩余质量; ② 以剩余合同与已有炉次为节点, 构造

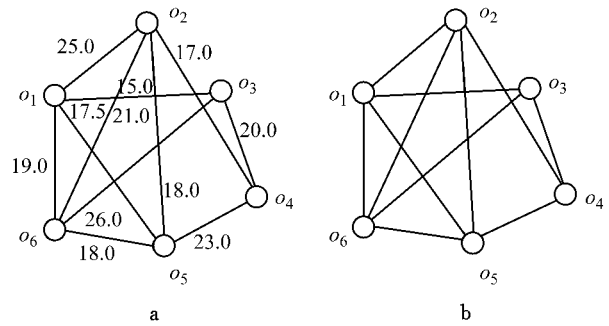


图 2 由合同之间构成的最大权重匹配非二分图
Fig. 2 Maximum weight non-bipartite matching between orders

一个如图 3 所示的二分图 G , 左侧节点为剩余合同 o_i , 右侧节点为构造的炉次 f_j . 如果合同 o_i 与炉次 j 兼容, 则增加一条边 $e=(i, j)$, 匹配重量为 w_{ij} . ③ 令 $R_{ij} = \min\{R_i, R_j\}$, 对于合同 o_i 与炉次 j , 令 $R_{ij} = \lfloor \frac{R_{ij}}{U_{i,\max}} \rfloor U_{i,\max}$, 合同 o_i 匹配到炉次 j 中的板坯数为 $U_{\text{num}} = \lfloor \frac{R_{ij}}{U_{i,\max}} \rfloor$, 板坯尺寸为 $U_{\text{size}} = \frac{R_{ij}}{U_{\text{num}}}$, 边 $e=(i, j)$ 的权重为 $w_{ij} = c'_{ij} \cdot R_{ij}$. ④ 在 G 中寻找一个最大加权二分匹配, 如果总的匹配为零, 则转步骤 3; 否则, 将 R_{ij} 质量的合同 o_i 匹配到第 j 个炉次中, 且令 $R_i = R_i - R_{ij}, R_j = R_j - R_{ij}$, 转回步骤 2 继续执行.

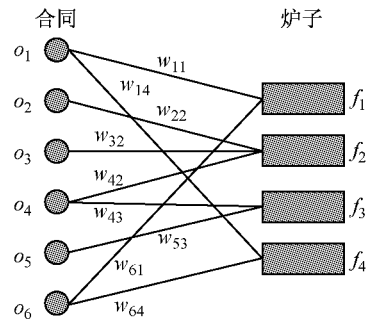


图 3 由合同与炉次构成的权重匹配二分图
Fig. 3 Weighted bipartite matching between order and furnace

步骤 3: 装箱算法. 该步骤模拟一维装箱过程, 将之前所获得的炉子视为箱子, 以剩余合同物件进行装箱. 步骤为, ① 根据步骤 2 得到的结果将剩余合同 o_i 划分成质量为 $U_{i,\min}$ 的板坯. ② 采用 Best-Fit 算法进行装箱, 具体步骤为令 $i=l$; 对合同 o_i , 依次从当前可用炉次集合中寻找一个成本最低的炉次, 即炉次钢级属于板坯主副钢级之一, 且炉次剩余质量 R_j 大于合同剩余质量; 令 $i=i+1$, 如果 $i < I$, 重复以上操作; 否则算法终止.

步骤 4: 调整板坯质量. 本步骤通过增加板坯大

小来进一步减少余材。由于步骤 2、步骤 3 板坯并不是依据最大坯重进行设计,故存在一定的调整空间。算法步骤为,①统计之前各合同所匹配的板坯数 $P_{number,i}$ 以及匹配质量 $W_{appweight,i}$;②构建最大流网络^[11]如图 4 所示。该网络为 4 层有向网络,输入层仅含一个输入节点,且与合同层所有节点相连,输出层仅含一个输出节点,炉次层所有节点均与输出节点相连。合同层由所有在之前步骤中参与匹配且未完成的合同组成,每一个节点代表一个合同。炉次层由步骤 1 中产生的炉子组成,每一个节点代表一个炉次。若某合同 o_i 有板坯匹配到某炉子 f_j ,则该合同对应节点与炉子对应节点相连。③设定网络中各炉容量。输入节点到第 i 个合同节点的弧的容量为 $P_{number,i} \cdot U_{imax} - W_{appweight,i}$;所有炉子节点到输出节点的弧容量为剩余炉容量 R_j ;设第 i 个合同匹配到第 j 个炉子的质量为 $W_{appweight,ij}$,板坯数 $P_{number,i}$,则第 i 个合同节点匹配到第 j 个炉子节点的弧的容量为 $P_{number,i} \cdot U_{imax} - W_{appweight,ij}$;④通过计算网络最大流,若第 i 个合同节点匹配到第 j 个炉子节点的弧的容量大于 $W_{appweight,ij}$,则可增加板坯质量。然后转步骤 5。

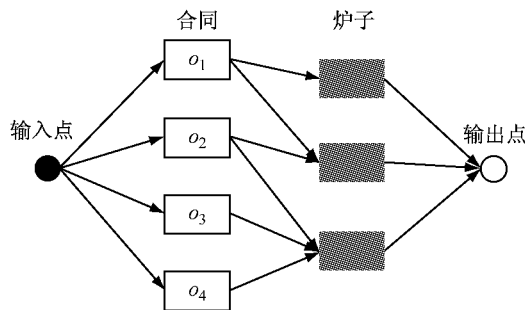


图 4 最大流网络
Fig.4 Maximum flow network

步骤 5:过滤筛选。依次检查匹配好的各炉,若某炉的余材量超过设定阈值,则将该炉内板坯重新放回合同,抛弃该炉,返回步骤 1。考虑到算法的收敛,算法过程中对阈值进行动态调整,逐步增大阈值。

3 实例计算

现以某钢厂实际生产数据为例,如表 1 所示共有 13 个合同,以第 1 个合同为例,若采用主钢种 1 生产,则其额外单位质量成本为零,若采用副钢种 3 生产,则其额外单位质量成本为 4。炼钢炉的容量上下限分别为 290t 与 310t。

启发式算法采用 C++ 编程实现,在 CPU 2.20

GHz、内存 1 GB 的配置下可在 2 s 内运算得到表 2 所示结果。表中第 5 列表示该炉的合同构成,如第 1 个炉次“1(28.5,2), 2(261.5,17)”表示该炉中包含第 1 个合同 28.5 t 共计 2 块板坯、第 2 个合同 261.5 t 共计 17 块板坯。从表 2 可知,第 3, 4, 5, 11, 12 炉中存在余材,共计 400 t,第 9 炉中合同 8 采用副钢种生产,共 13 t,额外生产成本为 26。

表 1 合同信息
Tab.1 Order information

合同序号	合同订货量/t	板坯质量/t	合同可选用的钢种	钢种替换成本
1	[220,240]	[14.1,15.5]	{1,2,3,4,5}	{0,2,4,6,8}
2	[310,340]	[13.3,15.7]	{1,2,3,4,5}	{0,2,4,6,8}
3	[160,190]	[12.8,13.6]	{2,3,4,6}	{0,2,4,8}
4	[230,250]	[14.5,15.9]	{2,3,4,6}	{0,2,4,8}
5	[155,170]	[16.4,18.7]	{3,4,5,7,8}	{0,2,4,8,10}
6	[370,395]	[15.1,16.5]	{4,5,6,7,8}	{0,2,4,6,8}
7	[320,345]	[13.8,14.3]	{4,5,6,7,8}	{0,2,4,6,8}
8	[240,270]	[13.0,13.5]	{4,5,6,7,8}	{0,2,4,6,8}
9	[180,200]	[14.3,15.7]	{5,6,7,8,9}	{0,2,4,6,8}
10	[190,200]	[14.6,16.1]	{5,6,7,8,9}	{0,2,4,6,8}
11	[210,230]	[16.0,17.4]	{5,6,7,8,9}	{0,2,4,6,8}
12	[190,210]	[15.8,16.4]	{7,8,9}	{0,2,4}
13	[210,230]	[13.9,15.1]	{7,8,9}	{0,2,4}

表 2 启发式算法得到的组炉优化结果
Tab.2 Charge optimization results by using heuristic method

炉次序号	炉次对应钢种	炉次质量	余材量	炉次合同
1	1	290.0	0	1(28.5,2), 2(261.5,17)
2	1	290.0	0	1(211.5,14), 2(78.5,5)
3	2	290.0	14.1	3(40.4,3), 4(235.5,16)
4	2	290.0	125.9	3(149.6,11), 4(14.5,1)
5	3	290.0	120.0	5(170.0,10)
6	4	310.0	0	6(60.4,4), 7(13.8,1), 8(235.8,18)
7	4	310.0	0	6(310.0,19)
8	4	306.2	0	7(306.2,22)
9	5	305.7	0	8(13.0,1), 9(180.0,12), 10(112.7,7)
10	5	303.5	0	10(77.3,5), 11(226.2,13)
11	7	290.0	87.8	13(202.2,14)
12	7	290.0	52.2	12(210.0,13), 13(27.8,2)

目前,现场较多采用的是基于规则的人工经验方法,原理如下:

(1)对所有合同以主钢级分类并以每类的主钢级进行组炉。如主钢级 G_i 类总质量为 w_i ,则可组成 $[w_i/W]$ 炉,其中, W 为一炉最大的生产量,剩余量为 R_i 。

(2)对剩余量 R_i 依可替代钢级进行合并,尽可能组成 1 炉,不够 1 炉的,仍以 1 炉生产,多余的为

余材。

表3为启发式算法与人工经验方法的优化结果对比,从中可以得到以下结论:

(1)针对相同的合同,启发式算法得到的板坯数量较少,说明单位板坯质量较大,减少了板坯切割次数和切割损失,提高了生产效率,且有利于产品运输。

(2)启发式算法可有效降低炉次数,减少了冶炼准备时间,提高了生产效率。

(3)启发式算法得到的组炉方案冶炼成本较低,说明大部分合同采用主钢级生产、少数合同采用“以优充次”的副钢级生产有效降低了生产成本。

(4)启发式算法可有效降低余材量,有利于减少库存,增加流动资金,从而提高经济效益。

(5)从计算效率来看,人工经验方法可在很短的时间内给出组炉方案,效率较高;启发式方法的计算时间在秒级,在实际生产中完全可以接受。

表3 启发式算法与人工经验方法的组炉优化结果对比

Tab.3 Comparison of charge optimization results between our method and artificial experience method

合同数	板坯数	炉次数	生产成本	余材	时间/s
13	210,226	12,13	26,104	400.0,636.9	2,1
20	331,346	19,21	52,130	422.4,919.6	3,1
30	489,505	28,30	69,182	433.7,926.5	4,1
45	752,764	43,46	95,216	465.2,1 206.0	7,2
56	894,915	50,54	121,252	483.4,1 439.6	10,2
69	1 094,1 115	59,63	147,286	497.7,1 492.4	13,3

注:逗号前为启发式算法,逗号后为人工经验方法。

从以上分析可知,启发式算法在优化效果上远远优于人工经验方法,在计算效率方面稍弱于人工经验方法,因此本文的方法能在较短的时间内得到好的组炉方案,对实际生产有指导意义。

4 结语

针对炼钢组炉计划编制问题,从合同出发,同时考虑板坯设计问题,建立了统一的多目标混合整数规划模型,并结合匹配算法、装箱算法及网络最大流算法设计出一套启发式算法。通过实际生产数据的

仿真研究表明,该算法能在较短的时间内给出合理的组炉方案,在减少余材量的同时减少了炼钢生产成本,有利于减少库存,提高生产效率。

参考文献:

- [1] Smith A, Smith B. Constraint programming approaches to a scheduling problem in steelmaking[C]//University of Leeds School of Computer studies Research Report Series. Leeds: Leeds School Press, 1997: 1-10.
- [2] Tang L X, Wang G S. Decision support system for the batching problems of steelmaking and continuous-casting production[J]. Omega, 2008, 36(6): 976.
- [3] 黄可为,卢克斌,汪定伟. 炼钢组炉问题优化模型及其动态规划算法[J]. 东北大学学报:自然科学版, 2006, 27(2): 138. HUANG Kewei, LU Kebin, WANG Dingwei. Dynamic programming algorithm and optimization model of charge design for steel-making [J]. Journal of Northeastern University: Natural Science, 2006, 27(2): 138.
- [4] 朱俊,贾树晋,杜斌. 基于PBIL与网络最大流的组炉算法[J]. 东北大学学报:自然科学版, 2012, 33(1): 52. ZHU Jun, JIA Shujin, DU Bin. PBIL and maximum-flow based algorithm of charge design problem [J]. Journal of Northeastern University: Natural Science, 2012, 33(1): 52.
- [5] 唐立新,杨自厚,王梦光. 炼钢-连铸最优炉次计划模型与算法[J]. 东北大学学报:自然科学版, 1996, 17(4): 440. TANG Lixin, YANG Zihou, WANG Mengguang. Model and algorithm of furnace charge plan for steelmaking-continuous casting production scheduling [J]. Journal of Northeastern University: Natural Science, 1996, 17(4): 440.
- [6] Heinz S, Schlechte T, Stephan R, et al. Solving steel mill slab design problems[J]. Constraints, 2012, 17: 39.
- [7] Dawande M, Kalagnanam J, Lee H S, et al. The slab-design problem in the steel industry[J]. Interface, 2004, 34(3): 215.
- [8] Stawowy A. Evolutionary based heuristic for bin packing problem[J]. Computer & Industrial Engineering, 2008, 55(2): 465.
- [9] Haouari M, Serairi M. Heuristics for the variable sized binpacking problem[J]. Computer & Operations Research, 2009, 36(10): 2877.
- [10] Kalagnanam J, Dawande M, Trumbo M, et al. The surplus inventory matching problem in the process industry [J]. Operations Research, 2000, 48(4): 505.
- [11] Ford L R, Fulkerson D R. Maximal flow through a network [J]. Canadian Journal of Mathematics, 1956, 8(1): 399.