

doi: 10.12068/j.issn.1005-3026.2017.07.001

# 求解具有时空约束的板坯库天车调度问题 Memetic 算法

王旭<sup>1</sup>, 刘士新<sup>1</sup>, 王佳<sup>2,3</sup>

(1. 东北大学 信息科学与工程学院, 辽宁 沈阳 110819; 2. 东北大学 工商管理学院, 辽宁 沈阳 110169;  
3. 东北大学秦皇岛分校 经济学院, 河北 秦皇岛 066004)

**摘 要:** 研究了钢铁企业板坯库天车调度问题, 考虑了时间和空间以及热轧计划等实际限制, 建立了一个板坯出库天车调度模型. 针对天车调度问题具有实时性和不可交叉性的特点, 设计了基于优先关系的天车分配编码方式、离散事件动态仿真解码、自适应交叉算子以及在交叉和变异后进行模拟退火局域搜索的 Memetic 算法. 通过某钢厂板坯出库过程中天车调度的实际数据对模型和算法进行仿真测试, 实验结果表明: 该算法具有很高的收敛性和稳定性, 满足实际生产需要.

**关 键 词:** 板坯库; 天车调度; Memetic 算法; 时空约束; 离散事件动态仿真

**中图分类号:** C 934      **文献标志码:** A      **文章编号:** 1005-3026(2017)07-0913-05

## Memetic Algorithm for Crane Scheduling Problem in Slab Yard with Spatial and Temporal Constraints

WANG Xu<sup>1</sup>, LIU Shi-xin<sup>1</sup>, WANG Jia<sup>2,3</sup>

(1. School of Information Science & Engineering, Northeastern University, Shenyang 110819, China; 2. School of Business Administration, Northeastern University, Shenyang 110169, China; 3. School of Economics, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China. Corresponding author: WANG Xu, E-mail: 45065706@qq.com)

**Abstract:** Crane scheduling problem was researched for slab yard in steel plant. Considering the time and space as well as the hot rolling, a crane scheduling model for slab yard was established. According to the characters of real-time and non-crossing for the problem, a Memetic algorithm was designed to solve it, including the crane allocation rule based on the priority relation, the decoding rules based on discrete event dynamic simulation (DEDS), adaptive crossover operator and simulate annealing global search after each crossover and mutation. Simulation experiment was performed with data by crane scheduling for slab out, and the results of simulation experiments showed that the proposed algorithm has high convergence speed and stability, which meets actual production demand.

**Key words:** slab yard; crane scheduling; Memetic algorithm; spatial and temporal constraints; discrete event dynamic simulation

板坯库天车调度属于多机多任务调度问题, 由于多台天车在同时执行多个任务, 且天车执行任务中受垛位布局、工艺流程、作业干涉等约束限制, 使板坯库天车调度问题成为典型的组合优化难题, 无法用一个有效的算法在多项式时间内求

出最优解<sup>[1]</sup>.

目前, 针对天车调度问题的研究主要集中在集装箱码头天车调度问题<sup>[2-4]</sup>和制造车间天车调度问题<sup>[5-6]</sup>, 而对冶金行业, 尤其是板坯库天车调度问题研究较少. 赵宁等<sup>[7]</sup>研究了板坯库天车作

收稿日期: 2016-02-04  
基金项目: 国家自然科学基金资助项目(61573089, 61333006, 71601040); 河北省高等学校社科研究基金资助项目(SQ162004).  
作者简介: 王旭(1982-), 男, 辽宁沈阳人, 东北大学博士研究生; 刘士新(1968-), 男, 辽宁调兵山人, 东北大学教授, 博士生导师.

业具有随机性的问题,建立了天车作业仿真模型,采用基于 Agent 技术的循环仿真方法对模型进行仿真优化,通过原始仿真模型和改进仿真模型的循环转换以及多档模糊评判寻求优化解.王旭等<sup>[8]</sup>研究了炼钢-连铸过程中天车调度问题,考虑时间和空间约束,建立了天车调度模型,采用 Memetic 算法对模型进行求解.Tang 等<sup>[9]</sup>研究了钢厂冷轧车间罩式退火的单天车调度问题,建立了以最小化钢卷堆垛时间为目标的混合整数规划模型,采用两阶段算法对模型进行求解.Tanizaki 等<sup>[10]</sup>研究了炼钢过程中多阶段天车调度问题,考虑天车作业干涉,建立了天车调度混合整数规划模型,采用启发式算法对模型进行求解.

以上研究成果通过为每个天车分配任务避免天车出现干扰,然而没有考虑出现干扰时天车避让情况.本文在已知轧制计划条件下,根据板坯库的垛位布局,考虑执行板坯出库任务中的板坯倒垛、天车避让等实际情况,建立以最小化板坯出库时间为目标的板坯库天车调度模型,设计了一个 Memetic 算法(Memetic algorithm,MA)对模型求解.

### 1 问题描述

如图 1 所示,在某板坯库中有  $K$  台天车,只能在作业跨内运行.沿作业跨方向,按照工作区域划分为  $M$  行,相邻两行距离为  $L$ .每行中有若干个板坯垛,同一行中板坯垛沿着作业跨的位置相同.天车作业是指天车从开始位置吊起板坯,运输到结束位置放下板坯的过程.为了避免天车发生碰撞,相邻天车之间必须保持一个最小的安全距离  $\delta$ .天车作业根据目标板坯所在板坯垛的位置划分为两种情况:如果目标板坯在顶层,天车可以将它直接运输到目的位置,称为运输作业.如果目标板坯上有阻碍板坯(见图 2,阴影为目标板坯),直到阻碍板坯被运到其他位置后,天车才可以将它运输到目的位置.清除阻碍板坯的过程称为倒垛作业.

本文天车调度问题描述如下:给定一个板坯出库计划  $\Omega$ ,包括  $N$  个作业任务,任务  $i(i=1,2,\cdots,N)$  有开始位置  $l_{si}$  和结束位置  $l_{ei}$ ,并且要在其时间窗  $[t_{ei},t_{li}]$  内完成.天车集合  $\Psi$ ,包括  $K$  台天车,天车  $k$  的起始位置  $l_k^0$ 、吊起和放下板坯的时间  $\mu$  以及运输板坯的速度  $v$  是预先给定的,且沿垂直作业跨方向的作业由天车上小车完成,可将小车运动时间归入天车作业时间.因此,任务  $i$  的处理时间  $t_i=2\cdot\mu+|l_{ei}-l_{si}|/v$ .类似地,任务  $i$  的结

束位置和任务  $j$  的开始位置之间的距离  $d_{ij}$ ,也可以预先计算出来.在任务开始时刻,指派天车执行任务,天车执行完任务后,从当前任务的结束位置以速度  $v$  空载到下一任务的开始位置.目标函数为最小化板坯出库的完成时间.

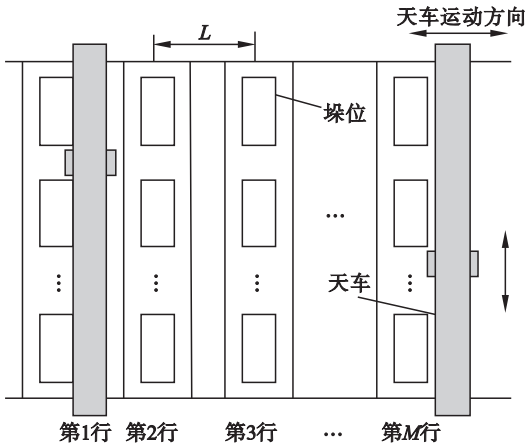


图 1 板坯库布局  
Fig. 1 Layout of slab yard

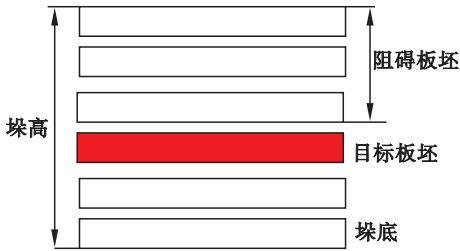


图 2 板坯垛结构  
Fig. 2 Structure of a slab stack

### 2 数学模型

为了清晰表达上面描述的问题,建立天车调度非线性整数规划模型.引入如下符号:

$\Omega$  为出库计划中任务集合,  $\Omega=\{1,2,\cdots,N\}$ ,  $N$  为任务数;  $\Gamma$  为任务之间优先级集合,  $(i,j)\in\Gamma$  表明任务  $i$  必须在任务  $j$  之前执行;  $\Psi$  为天车集合,按照从左到右的顺序依次进行编号,  $\Psi=\{1,2,\cdots,K\}$ ;  $K$  为天车数;  $l_k^0$  为天车  $k$  的开始位置;  $l_{si}$  为任务  $i$  的开始位置;  $l_{ei}$  为任务  $i$  的结束位置;  $t_{ei}$  为任务  $i$  的最早开始时间;  $t_{li}$  为任务  $i$  的最晚结束时间;  $t_i$  为任务  $i$  的处理时间(包括天车吊起、运行和放下板坯的时间);  $v$  为天车运行速度;  $\delta$  为天车之间的最小距离,即需要保持的安全距离;  $d_{ij}$  为天车从任务  $i$  的结束位置到任务  $j$  的开始位置的距离;  $T_{\max}$  为计划展望期;  $M$  为足够大的正数.

决策变量:  $t_{fi}$  为任务  $i$  的结束时间;  $x_k(t)$  为天

车  $k$  在时刻  $t$  的位置.

$$y_{ijk} = \begin{cases} 1, & \text{如果天车 } k \text{ 执行任务 } i \text{ 之后执行任务 } j; \\ 0, & \text{否则.} \end{cases}$$

建立数学模型如下:

$$\min \max_{i \in \Omega} (t_{fi}). \quad (1)$$

$$\text{s. t.} \quad \sum_{k \in \psi} \sum_{i \in \Omega} y_{ijk} = 1, \forall j \in \Omega. \quad (2)$$

$$\sum_{k \in \psi} \sum_{j \in \Omega} y_{ijk} = 1, \forall i \in \Omega. \quad (3)$$

$$x_k(t) - v \cdot \Delta t \leq x_k(t + \Delta t) \leq x_k(t) + v \cdot \Delta t, \forall k \in \psi, \\ t = 1, 2, \dots, T_{\max}. \quad (4)$$

$$x_k(t) \leq x_{k+1}(t) - \delta, \forall k \in \psi, t = 1, 2, \dots, T_{\max}. \quad (5)$$

$$t_{fi} + t_j + \frac{d_{ij}}{v} \leq t_{fj} + M \cdot (1 - y_{ijk}), \forall i, j \in \Omega \text{ 且 } i \neq j, \\ \forall k \in \psi. \quad (6)$$

$$t_{ei} + t_i \leq t_{fi} \leq t_{li}, \forall i \in \Omega. \quad (7)$$

$$t_{fi} + t_j + \frac{d_{ij}}{v} \leq t_{fj}, \forall (i, j) \in \Gamma. \quad (8)$$

$$X_{ijk} \in \{0, 1\}, \forall i, j \in \Omega, \forall k \in \psi. \quad (9)$$

其中,式(1)为目标函数,即最小化板坯出库时间;式(2)和式(3)每个任务只能由一台天车执行一次.式(4)为天车运行速度约束;式(5)表示两台天车需保持一定安全距离;式(6)和式(7)表示任务的时间窗约束;式(8)表示任务  $i$  要在任务  $j$  之前完成;式(9)定义变量的取值范围.

### 3 MA 设计

本文 MA 分别采用遗传算法和模拟退火算法作为群体进化方法和局域搜索方法.具体步骤如下:

1) 参数初始化:种群规模  $P$ ,迭代代数  $G$ ,当前最优个体  $I_{\text{best}}$ ,相同最好解出现次数  $n_{\text{best}}$ ,保持前代最好解数量  $n_{f,\text{best}}$ ,初始温度  $\theta_{\max}$ ,终止温度  $\theta_{\min}$ ,温降系数  $c$ ,温降幅度  $\Delta t$ ;

2) 初始化种群  $P(1)$ ,令  $P'(1) := \emptyset$ ;

For  $k = 1$  To  $G$

For  $i = 1$  To  $P/2$

采用轮盘赌方式从  $P(k)$  选择两个个体  $S_1$  和  $S_2$ ;

应用自适应交叉算子从  $S_1$  和  $S_2$  构造两个新个体  $S'_1$  和  $S'_2$ ;

应用变异算子从  $S'_1$  和  $S'_2$  构造两个新个体  $S''_1$  和  $S''_2$ ;

对  $S''_1$  和  $S''_2$  分别应用 SA 局域搜索得到个体  $S^*_1$  和  $S^*_2$ ;

$$P'(k) := P(k) \cup \{S^*_1, S^*_2\};$$

从  $P'(k)$  中选择最好个体作为  $I_{\text{best}}$ ,从  $P(k) \cup P'(k)$  中选择最好的  $n_{f,\text{best}}$  个个体,从  $P(k) \cup P'(k)$  中采用轮盘赌方式选择  $(P - n_{f,\text{best}})$  个个体,生成  $P''(k)$ ;

If 相同  $I_{\text{best}}$  出现次数  $\geq n_{\text{best}}$ , 结束,输出  $I_{\text{best}}$ ;

Else  $P(k+1) := P''(k)$ ;

3) Return 输出当代种群最好个体.

#### 3.1 编码和解码

个体  $I$  编码采用长度为  $N$  的单链编码方式,如图 3 所示,其中  $N$  为任务总数,位置  $i$  为第  $i$  个执行的任务,  $C_i$  为执行任务  $i$  的天车编号.解码过程就是根据  $I$  编码,采用离散事件动态仿真方法执行任务的过程.

|       |       |         |       |         |       |
|-------|-------|---------|-------|---------|-------|
| $C_1$ | $C_2$ | $\dots$ | $C_i$ | $\dots$ | $C_N$ |
|-------|-------|---------|-------|---------|-------|

图3 个体编码

Fig. 3 Individual coding

#### 3.2 初始种群的产生

初始种群生成步骤如下:①令  $i = 0$ ,生成  $[1, K]$  区间的随机整数,填入位置  $i$ ;②令  $i = i + 1$ ,重复①,直到满足  $i = N$  为止;③重复①~②直到  $P$  次,生成初始种群.

#### 3.3 交叉算子

采用概率为  $P_c$  的两点交叉算子对种群中两个个体进行交叉操作.为了增加种群多样性,根据式(10)自适应地调整  $P_c$ ,控制种群中交叉次数.

$$P_c = \begin{cases} P_c \times \frac{f_{\max} - f}{f_{\max} - f_{\min}}, & f < f_{\text{avg}}; \\ P_c, & f \geq f_{\text{avg}}. \end{cases} \quad (10)$$

其中: $f$ 为每次交叉个体最大目标函数值; $f_{\max}$ 为种群最大目标函数值; $f_{\text{avg}}$ 为种群平均目标函数值; $f_{\min}$ 为种群最小目标函数值.交叉算子描述如下:①生成区间  $[1, N]$  的随机数  $i$  和  $j$  ( $i \neq j$ );②交换父代个体  $S_1$  和  $S_2$  中区间  $[i, j]$  上的基因,其他位置上基因保持不变,生成子代个体  $S'_1$  和  $S'_2$ .

#### 3.4 变异算子

采用概率为  $P_m$  的变异算子对种群中个体进行变异操作.变异算子描述如下:①生成区间  $[0, N]$  的随机整数  $i$  和  $j$  ( $i \neq j$ );②交换父代个体  $S'$  中位置  $i$  和  $j$  上的基因,生成子代个体  $S''$ .

#### 3.5 局域搜索算子

本文提出一种改进的模拟退火局域搜索算法,描述如下:①初始化初始温度  $\theta_{\max}$ ,截止温度  $\theta_{\min}$ ,降温幅度  $\Delta\theta$ ,降温系数  $c$ .②令交叉变异次

数  $i=0$ , 当前温度  $\theta_i = \theta_{\max}$ . ③计算  $\Delta = f(i+1) - f(i)$ , 其中  $f(i)$  为第  $i$  次交叉和变异后种群中最好个体的目标函数值. 若  $\Delta < 0$  或  $\Delta > 0$  且  $\exp(-\Delta/\theta_i) > \text{random}(0,1)$ , 接受第  $i$  次交叉和变异, 终止算法; 否则, 重新进行交叉变异. ④令  $i = i+1$ , 改变退火温度, 使  $\theta_{i+1} = \theta_i - \Delta\theta_i$ . 其中,  $\Delta\theta_i = c\Delta\theta$ , 当  $\theta_i > \frac{2}{5} \times (\theta_{\max} + \theta_{\min})$  时,  $c = 1$ ; 当  $t_i \leq \frac{2}{5} \times (\theta_{\max} + \theta_{\min})$  时,  $c = 0.8$ , 返回步骤③. ⑤ $\theta_i < \theta_{\min}$ , 终止算法.

### 4 算例设计及结果分析

本文第 3 节的 MA 应用 C++ 语言编写, 运行于 Windows7 操作系统的 Core2/2.67 GHz 内存 4GB 的 HP 兼容机上. 为了测试本文算法的性能, 设计了遗传算法 (genetic algorithm, GA) 进行对比.

#### 4.1 算例设计

板坯库内按照工作区域划分为 6 行, 相邻两行距离为 20 m, 每行中有 3 个板坯垛. 有 2~3 台天车在作业跨上工作, 天车的初始位置随机分配. 天车运行速度为 10 m/min, 安全距离为 20 m. 为了测试算法的有效性, 取国内某钢铁厂热轧计划下 8 组不同规模的板坯出库实际数据对算法进行测试, 每组算例参数见表 1.

#### 4.2 对照算法

GA 采用 3.1 节介绍的编码和解码方式, 轮盘赌选择, 采用单点交叉算子和单点变异算子. 参数设置如下: 种群规模  $P=20$ , 迭代数  $G=20$ , 交叉率  $P_c=0.85$ , 变异率  $P_m=0.01$ .

| 表 1 算例参数设置<br>Table 1 Parameters setting |     |     |     |         |
|--|-----|-----|-----|---------|
| 算例                                       | 天车数 | 板坯数 | 任务数 | 优先关系任务数 |
| 1  | 2   | 20  | 28  | 8       |
| 2  | 3   | 20  | 28  | 8       |
| 3  | 2   | 30  | 57  | 26      |
| 4  | 3   | 30  | 57  | 26      |
| 5  | 2   | 40  | 79  | 40      |
| 6  | 3   | 40  | 79  | 40      |
| 7  | 2   | 50  | 91  | 40      |
| 8  | 3   | 50  | 91  | 40      |

#### 4.3 算法对比分析

本文 MA 参数设置如下: 种群规模  $P=20$ , 迭

代数  $G=20$ , 交叉率  $P_c=0.85$ , 变异率  $P_m=0.01$ , 初始温度  $\theta_{\max}=80\text{ }^{\circ}\text{C}$ , 截止温度  $\theta_{\min}=40\text{ }^{\circ}\text{C}$ , 降温幅度  $\Delta\theta=20\text{ }^{\circ}\text{C}$ , 降温系数  $c=1$ .

采用 GA 和 MA 分别对 8 组算例进行求解, 每组算例运行 5 次, 取其中最优结果. 统计结果如表 2 所示, 其中,  $T$  表示算法运行时间,  $O$  表示目标函数值.

| 表 2 两种算法实验结果对比<br>Table 2 Comparison of results of two algorithms |                |       |                |       |           |           |
|---|----------------|-------|----------------|-------|-----------|-----------|
| 算例  | GA             |       | MA             |       | $T_1/T_2$ | $O_1/O_2$ |
|   | $T_1/\text{s}$ | $O_1$ | $T_2/\text{s}$ | $O_2$ |           |           |
| 1   | 0.73           | 65    | 0.74           | 65    | 0.99      | 1.00      |
| 2   | 1.32           | 60    | 1.46           | 59    | 0.90      | 1.02      |
| 3   | 3.18           | 82    | 3.28           | 81    | 0.97      | 1.01      |
| 4   | 5.51           | 76    | 6.23           | 74    | 0.88      | 1.03      |
| 5   | 7.41           | 129   | 7.56           | 127   | 0.98      | 1.01      |
| 6   | 11.36          | 123   | 13.51          | 119   | 0.84      | 1.03      |
| 7   | 6.38           | 93    | 6.75           | 92    | 0.95      | 1.02      |
| 8   | 7.52           | 88    | 8.67           | 85    | 0.87      | 1.04      |

从表 2 可以看出, 两种算法均能在较短的时间内求得近优解, 随着问题规模的增大, 两种算法的求解时间相应增大. GA 的求解时间均比 MA 的求解时间短, 但 GA 的求解结果均差于 MA 的求解结果. 通过对比不同规模算例的求解结果得出结论: ①在天车数量保持不变的情况下, 增加任务数量, 板坯出库时间增大, 二者呈线性变化. ②在任务数保持不变的情况下, 增加天车数量可以降低板坯出库时间, 但降幅较小. 因为过多天车导致空载增加, 干扰现象增多. ③随着优先关系任务数与板坯数的比值增加, 导致每块板坯出库时, 天车执行倒垛任务次数增加, 造成板坯出库时间增长. ④相比于 GA, MA 在个体搜索后, 嵌入局域搜索, 得到更好解的几率增大. ⑤相比于 GA, MA 虽然每次迭代中搜索了更多的空间, 但同时也付出了更大的时间代价.

如图 4 所示, 从最优值比较可以看出, GA 曲线走势平缓, 经过几次小幅下降后收敛到最好解. 而 MA 走势陡峭, 经过几次大幅下降之后, 才收敛到最好解. 这是因为 GA 只有群体进化, 搜索空间有限, 容易陷入局部最优, 而 MA 在群体进化后都有局域搜索, 扩展搜索空间, 跳出局部最优的能力增强. 从平均值比较可以看出, GA 前期一直处于往复震荡状态, 最后趋于稳定. 而 MA 一直处于平稳下降状态, 最后趋于稳定. 这是因为 GA 中群体进化的盲目性, 没有一个定向搜索方向. 而 MA 在群体进化后都有局域搜索, 在种群中最优



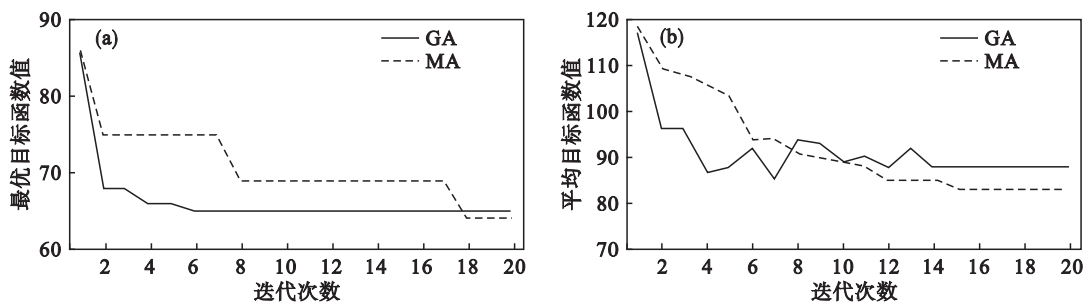


图 4 每次迭代最优值比较和平均值比较  
Fig. 4 Comparison of the optimal values and the mean values for each iteration  
(a)—最优目标函数值；(b)—平均目标函数值。

个体牵引作用下,由随机搜索转化为定向搜索.

图 5 为两种算法求解算例 1 时每次迭代运行时间比较. MA 每次迭代的时间均比 GA 长,导致最后收敛时间比 GA 长. 这是因为 MA 算法每次迭代都要进行局域搜索,导致目标函数比较次数增多,增加算法运行时间.

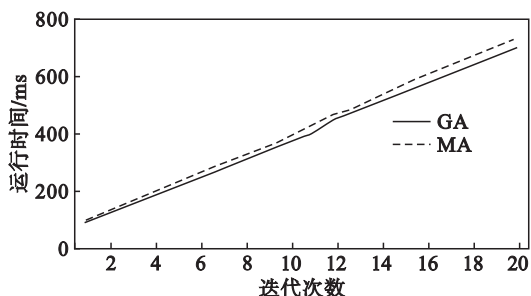


图 5 每次迭代运行时间比较  
Fig. 5 Comparison of the running time for each iteration

## 5 结 论

本文针对钢铁厂板坯库天车调度问题,考虑了时空约束以及热轧计划的限制,建立了天车调度优化模型,采用 MA 算法对模型进行求解. 针对问题的特点,设计了编码和解码规则、自适应交叉算子、变异算子和模拟退火局域搜索算子. 通过板坯库天车调度实际数据对模型和算法进行了测试,并与 GA 的求解结果进行对比,证明该算法是求解板坯库天车调度问题的一种有效算法.

## 参考文献:

[1] Dohn A, Clausen J. Optimizing the slab yard planning and crane scheduling problem using a two-stage heuristic [J]. *International Journal of Production Research*, 2010, 48(15): 4585 – 4608.

[2] Matsuo H, Shang J S, Sullivan R S. A crane scheduling problem in a computer-integrated manufacturing environment [J]. *Management Science*, 1991, 37(5): 587 – 606.

[3] Kim K H, Park Y M. A crane scheduling method for port container terminals [J]. *European Journal of Operation Research*, 2004, 156(3): 752 – 768.

[4] Jung S H, Kim K H. Load scheduling for multiple quay cranes in port container terminals [J]. *Journal of Intelligent Manufacturing*, 2006, 17(4): 479 – 492.

[5] Tamaki H, Murao H. Simulation-based optimization model and meta heuristic solution of multiple crane scheduling problem [C]// *IEEE International Conference on Systems, Man and Cybernetic*. Shanghai, 2004: 1469 – 1472.

[6] Liu P, Tang L X. The refining scheduling problem with crane non-collision constraint in steelmaking process [C]// *IEEE International Conference on Automation and Logistics*. Qingdao, 2008: 536 – 541.

[7] 赵宁, 杜彦华, 董绍华. 基于循环仿真的钢铁板坯库天车作业优化 [J]. *系统工程理论与实践*, 2012, 32(12): 2925 – 2830.

(Zhao Ning, Du Yan-hua, Dong Shao-hua. Optimization of crane scheduling in slab yard based on cycle simulation [J]. *Systems Engineering—Theory & Practice*, 2012, 32(12): 2925 – 2830.)

[8] 王旭, 刘士新, 王佳. 求解具有时空约束的天车调度问题 Memetic 算法 [J]. *东北大学学报 (自然科学版)*, 2014, 35(2): 191 – 194.

(Wang Xu, Liu Shi-xin, Wang Jia. Memetic algorithm for crane scheduling problem with spatial and temporal constraints [J]. *Journal of Northeastern University (Natural Science)*, 2014, 35(2): 191 – 194.)

[9] Tang L X, Xie X. Scheduling of a single in batch annealing process [J]. *Computers and Operation Research*, 2009, 36(10): 2835 – 2865.

[10] Tanizaki T, Tamura T, Sakai H. A heuristic scheduling algorithm for steel making process with crane handling [J]. *Journal of the Operations Research Society of Japan*, 2006, 49(3): 188 – 201.