

# 热轧带钢轧制批量计划优化模型及算法

刘士新<sup>1</sup>, 宋健海<sup>2</sup>, 周山长<sup>1</sup>

(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 110004;

2. 上海宝信软件股份有限公司 MES事业部, 上海 201900)

**摘要:** 基于奖金收集车辆路径问题模型建立了热轧带钢生产批量计划多目标优化模型. 模型综合考虑了生产工艺约束、用户合同需求以及综合生产指标优化等因素. 利用加权函数法将多目标优化模型转换为单目标优化模型. 针对模型特点设计了蚁群优化求解算法, 算法中嵌入了单向插入和2-opt局部搜索过程. 引用某钢铁企业热轧生产轧制批量计划编制的实际问题对模型和算法进行了验证, 结果表明模型和算法的优化效果和时间效率是令人满意的.

**关键词:** 轧制批量计划; 奖金收集车辆路径问题; 多目标; 蚁群最优化; 局部搜索过程

**中图分类号:** C934      **文献标识码:** A

## Model and algorithm for solving hot strip rolling batch planning problems

LIU Shi-xin<sup>1</sup>, SONG Jian-hai<sup>2</sup>, ZHOU Shan-chang<sup>1</sup>

(1. School of Information Science & Engineering, Northeastern University; Key Laboratory of Process Industry Automation, Ministry of Education, Shenyang Liaoning 110004, China;

2. MES Business Department, Shanghai Baosight Software Limited Company, Shanghai 201900, China)

**Abstract:** A prize-collecting-vehicle-routing-problem (PCVRP) based multi-objective model is formulated for solving hot rolling batch planning problems in this paper. Firstly, the hot strip rolling batch planning is modelled by considering the production process program constraints, users' requirements and production integration objectives. Weighted-sum approach is then used to transform the multi-objective model into a single objective one. An ant colony optimization (ACO) algorithm embedded with single-direction-insert and 2-opt local search procedures is also designed to solve it. Finally, taking a practical hot strip rolling lot planning problem as instance, the model and algorithm are tested for effectiveness and efficiency. Computational results show that the model and algorithm are satisfactory.

**Key words:** hot strip rolling batch planning; PCVRP (prize-collecting-vehicle-routing-problem); multi-objective; ACO (ant colony optimization); local search procedure

### 1 引言(Introduction)

热轧带钢是钢铁企业产品的重要组成部分, 热轧带钢生产管理的一个主要任务是编制轧制批量计划. 由于热轧带钢生产过程的高度复杂性, 轧制计划的好坏直接影响产品质量、轧辊成本和生产效率等. 因此, 轧制计划优化问题一直吸引着众多业者和学者的研究和关注. 目前研究主要分为轧制单元计划优化和轧制批量计划优化两个方面. 在轧制单元计划优化方面, 主要是基于旅行商问题模型建立热轧单元计划优化模型<sup>[1~4]</sup>. 在轧制批量计划优化方面, 主要基于车辆路径问题模型(vehicle routing problem, VRP)<sup>[5~7]</sup>和多旅行商问题模型建立

优化模型<sup>[8]</sup>. 本文基于奖金收集车辆路径问题(prize-collecting-vehicle-routing-problem, PCVRP)模型建立了热轧带钢轧制批量计划多目标优化模型, 设计了蚁群最优化(ant colony optimization, ACO)<sup>[9,10]</sup>算法进行求解.

### 2 数学模型(Mathematical model)

热轧带钢生产以板坯为原材料, 经过加热、粗轧、精轧、层流冷却、卷取和精整等工序加工后形成热轧钢卷. 热轧带钢的产品质量主要由精轧机组连轧时保证. 精轧机组一般由6~7个机架组成, 每个机架配置上下两个工作辊和两个支撑辊. 由于高温、高速轧制, 轧辊磨损很大, 为保证带钢的质量, 需要

定期更换. 更换两个工作辊之间轧制的板坯称为一个轧制单元, 多个轧制单元构成一个轧制批量.

一个完整的轧制单元由烫辊材和主体材两部分组成, 单元中的板坯编排在宽度上呈“双梯形结构”<sup>[3]</sup>. 烫辊材主要用于加热轧辊, 所需板坯数量少、产品质量要求低、容易编制, 因此, 不在本文优化范围之内.

主体材板坯是轧制单元的主要组成部分, 编制计划时通常要满足以下规程: 1) 板坯在硬度和厚度上要求平稳变化; 2) 带钢宽度总体上呈递减变化(允许局部递增变化), 并且具有相同宽度的带钢连续轧制长度不能超过某一限制; 3) 整个轧制单元总轧制长度应小于轧辊最大轧制长度.

热轧带钢轧制批量计划优化过程就是根据轧制规程约束, 对原料板坯进行选择、聚类 and 排序的过程. 选择过程就是根据合同交货期和合同板坯特性来选定要编入计划的板坯; 聚类过程就是将选定的原料板坯分类成若干个轧制单元; 排序过程则是对每个轧制单元内的板坯, 根据板坯属性进行排序, 以减少轧辊磨损、提高产品质量和生产效率. 在整个优化过程中, 板坯的宽度、厚度、硬度以及合同交货期等特性是需要重点考虑的因素.

热轧带钢的轧制批量计划对于产品质量、作业成本、生产效率都起到非常关键的作用. 计划编制过程要追求高效率、高质量、准时交货、低成本、低能耗等多个互相矛盾的目标, 是一个典型的多目标优化问题. 本文针对热轧带钢轧制批量计划问题建立了多目标PCVRP模型.

PCVRP是VRP<sup>[11,12]</sup>的扩展, 与标准VRP不同的是: PCVRP中的每个节点被赋以一定的奖励值和惩罚值, 如果某节点被访问到, 可以得到与该节点对应的奖励, 否则会受到相应的惩罚. 问题是如何设计车辆行使路线使得用最少的车辆服务若干个顾客, 每辆被调用车辆获得奖励大于某一预设值, 且所有被调用车辆总行使费用加上因未访问节点受到的惩罚总和最小.

将板坯看作PCVRP中的客户, 引入如下符号, 可基于PCVRP建立轧制批量计划优化模型. 记 $V$ 为全部板坯集合,  $i \in V$ 对应板坯 $i$ ,  $n = |V|$ 为板坯总数,  $i = 0$ 对应虚拟板坯, 板坯 $i$ 的轧制长度 $l_i$ 对应客户对车辆的能力需求.  $K$ 为轧制批量计划中包含的轧制单元数量.  $V_k \subseteq V$ 为第 $k$ ,  $k = 1, \dots, K$ 个轧制单元包含的板坯序列.  $d_{ij}^w, d_{ij}^t, d_{ij}^h$ 分别为轧制单元内相邻板坯间宽度、厚度和硬度跳跃惩罚, 则边 $(i, j)$ 的权重 $d_{ij} = d_{ij}^w + d_{ij}^t + d_{ij}^h$ , 对应轧制单元内板坯顺序为 $i \rightarrow j$ 时的宽度、厚度和硬度跳跃惩罚总和, 其中,  $d_{0i} = d_{i0} = 0, i = 1, 2, \dots, n$ .  $p_i$ 为板坯 $i$ 被编

入计划时获得的奖金, 其值被设置为板坯 $i$ 的轧制长度 $l_i$ .  $q_i$ 为板坯 $i$ 未被编入轧制计划时受到的惩罚, 其值被设置为板坯轧制长度乘以合同交货期优先系数,  $q_0 = \infty$ .  $B$ 为奖金收集下限, 对应每个轧制单元的最小轧制长度.  $Q$ 为车辆能力, 对应每个轧制单元的最大轧制长度.  $\gamma_{ik}$ 为第 $k$ 个轧制单元内与板坯 $i$ 连续同宽轧制的板坯集合.  $R$ 为每个轧制单元内同宽板坯连续轧制长度限制.  $\lambda$ 为轧制单元间设备调整费用.  $x_{ijk} = 1$ , 如果在第 $k$ 个轧制单元内板坯 $j$ 紧接在板坯 $i$ 后轧制, 否则 $x_{ijk} = 0$ .  $y_{ik} = 1$ , 如果板坯 $i$ 被编入第 $k$ 个轧制单元, 否则 $y_{ik} = 0$ . 则热轧轧制批量计划多目标优化模型描述如下:

$$\min Z_1 = \sum_{k=1}^K \sum_{i \in V_k} \sum_{j \in V_k \setminus \{i\}} d_{ij} \cdot x_{ijk}, \quad (1)$$

$$\min Z_2 = \sum_{i=1}^n q_i \cdot (1 - \sum_{k=1}^K y_{ik}), \quad (2)$$

$$\min Z_3 = \lambda \cdot \sum_{k=1}^K \min(1, \sum_{i=1}^n y_{ik}), \quad (3)$$

s.t.

$$\sum_{j \in V_k \setminus \{i\}} x_{ijk} = y_{ik}, \quad i \in V_k, \quad k = 1, \dots, K, \quad (4)$$

$$\sum_{i \in V_k \setminus \{j\}} x_{ijk} = y_{jk}, \quad j \in V_k, \quad k = 1, \dots, K, \quad (5)$$

$$y_{0k} = 1, \quad k = 1, \dots, K, \quad (6)$$

$$\sum_{k=1}^K y_{ik} \leq 1, \quad i = 1, \dots, n, \quad (7)$$

$$\sum_{i \in V_k} p_i \cdot y_{ik} \geq B, \quad k = 1, \dots, K, \quad (8)$$

$$\sum_{i \in V_k} p_i \cdot y_{ik} \leq Q, \quad k = 1, \dots, K, \quad (9)$$

$$\sum_{j \in \gamma_{ik}} l_j \leq R, \quad i \in V_k, \quad k = 1, \dots, K, \quad (10)$$

$$\sum_{i, j \in V_k} x_{ijk} \leq |V_k| - 1, \quad k = 1, \dots, K. \quad (11)$$

其中: 目标函数(1)代表板坯厚度、硬度和宽度跳跃惩罚最小; 目标函数(2)代表未编入轧制计划板坯总惩罚最小; 目标函数(3)代表轧制计划内各单元设备调整费用总和最小; 公式(4)表示在已安排的轧制计划中, 板坯 $i$ 后有且只有一块板坯; 公式(5)表示在已安排的轧制计划中, 板坯 $j$ 前有且只有一块板坯; 公式(6)表示虚拟板坯0一定被安排到每个轧制单元中; 公式(7)表示板坯 $i$ 最多只能被安排到一个轧制单元中; 公式(8)表示每个轧制单元计划的最小轧制长度限制; 公式(9)表示每个轧制单元计划的最大的轧制长度限制; 公式(10)表示每个轧制单元计划内同宽板坯轧制长度限制; 公式(11)避免出现子回路.

模型(1)~(11)属于多目标非线性0-1优化模型. 本文采用线性加权法处理多目标优化函

数. 记 $w_1, w_2$ 和 $w_3$ 分别是目标函数(1)~(3)的权重系数, 则目标函数(1)~(3)可以转换成以下形式:

$$\min Z = w_1 \times Z_1 + w_2 \times Z_2 + w_3 \times Z_3. \quad (12)$$

实际应用中的轧制批量计划优化问题多属于大规模组合优化问题, 求解困难. 本文针对模型特点设计了ACO求解算法, 利用蚂蚁的行走过程实现可行解的构造过程. 在算法中, 给蚂蚁赋予一定智能, 使其能够识别解的构造过程的约束满足情况, 并利用蚂蚁群的协同机制实现解的进化过程.

### 3 优化算法(Optimizing algorithm)

#### 3.1 ACO算法(ACO algorithm)

鉴于ACO算法在TSP以及其他组合优化问题中的成功应用<sup>[9,13]</sup>, 本文设计了ACO算法用来求解模型(12)(4)~(11). 算法中, 板坯作为蚂蚁要访问的节点, 板坯之间的接续关系作为节点之间的边, 边 $(i, j)$ 的权重为 $d_{ij}$ . 如果相邻两块板坯因具有相同的厚度、硬度和宽度而跳跃总惩罚值为0, 则将跳跃总惩罚值设置为较小的值0.01. 求解过程中每个蚂蚁按照如下方法构造旅行路径: 从虚拟板坯0出发, 从可选板坯集合中选择具有最大宽度的某一块板坯作为第1个轧制单元主体材的第1块板坯, 然后根据边上信息素浓度 $\tau_{ij}$ 、启发优先系数 $\eta_{ij}$ 以及约束条件随机选择下一块板坯, 如果该板坯不满足连续同宽轧制长度限制的约束条件, 则将该板坯加入禁忌表Tabu中, 否则将该板坯加入解集合. 蚂蚁在获得的奖金达到 $B$ 后以概率 $P$ 返回起始节点0, 如果由于轧制能力的限制而不能继续轧制其它板坯时, 则必须返虚拟板坯0. 然后释放禁忌表Tabu, 开始新轧制单元的构造过程. 重复以上轧制单元的构造过程, 直到剩余板坯的轧制长度总和小于轧制单元最小轧制长度为止完成一次解的构造过程.

为了更好地描述本文的ACO算法, 引入以下符号: 记 $m$ 为系统中蚂蚁的数量;  $k$ 为算法中蚂蚁的编号;  $kl$ 为蚂蚁 $k$ 当前所处的节点;  $\tau_{ij}$ 为某一求解状态下边 $(i, j)$ 上的信息素浓度;  $\eta_{ij}$ 为某一求解状态下蚂蚁由节点 $i$ 转移到 $j$ 的启发优先系数, 当蚂蚁处于节点 $i(i \neq 0)$ 时,  $\eta_{ij} = 1/d_{ij}$ ;  $M_k(i)$ 为处于节点 $i$ 的蚂蚁 $k$ 下一步可以访问的节点集合, 根据问题约束确定;  $\alpha$ 为信息素浓度 $\tau_{ij}$ 的相对重要性度量;  $\beta$ 为启发优先系数 $\eta_{ij}$ 的相对重要性度量;  $\rho$ 为信息素挥发系数;  $k_{gb}$ 为从算法开始执行到目前为止产生了最好解的蚂蚁编号;  $P$ 为蚂蚁在获得奖金数量达到 $B$ 后返回起始节点0的概率. 则在算法中, 处于节点0的蚂蚁 $k$ 从 $M_k(i)$ 中选取板坯宽度最大的某块板坯作为主体材的第1块板坯, 而处于节点 $i(i \neq 0)$ 的蚂蚁 $k$ 依概率 $p_k(i, j)$ 选择下一节点 $j$ ,

$$p_k(i, j) = \begin{cases} \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{z \in M_k(i)} [\tau_{iz}]^\alpha \cdot [\eta_{iz}]^\beta}, & j \in M_k(i), \\ 0, & \text{其他.} \end{cases} \quad (13)$$

在算法迭代过程中,  $\tau_{ij}$ 的值依据式(14)进行延迟更新:

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \Delta\tau_{ij}. \quad (14)$$

其中 $\Delta\tau_{ij}$ 为信息素更新量, 在每次迭代完成后, 按式(15)计算

$$\Delta\tau_{ij} = \begin{cases} W/f_{gb}, & \text{如果}(i, j) \text{包含在蚂蚁} k_{gb} \text{的路径中,} \\ 0, & \text{其他.} \end{cases} \quad (15)$$

式中:  $W$ 为常数,  $f_{gb}$ 为蚂蚁 $k_{gb}$ 产生的解的目标函数值, 根据式(12)计算.

本文的ACO算法求解步骤如过程1所示. 其中,  $T_k$ 为第 $k$ 个蚂蚁经过的节点集合;  $T'_k$ 为蚂蚁 $k$ 当前正在构造的轧制单元计划的主体材板坯集合,  $T'_k \subseteq T_k$ ;  $T_{gb}$ 为蚂蚁 $k_{gb}$ 经过的节点集合. 未说明的符号意义同前. 步骤5的终止条件为完成 $\max Iter$ 次循环.

#### 过程1

1° 初始化:  $\tau_{ij} := \tau_0, i, j = 0, 1, \dots, n, i \neq j; T_{gb} := \Phi;$

2° 设置每个蚂蚁的出发节点为0,  $T_k := \{0\}, k = 1, \dots, m;$

For  $k := 1$  To  $m$  Do

从当前可选板坯中选取宽度最大的某块板坯作为蚂蚁 $k$ 访问的第1块板坯;

End-For

For  $k := 1$  To  $m$

finish := false

Do

更新 $T'_k$ , 计算蚂蚁 $k$ 当前正在构造的轧制单元主体材轧制长度 $CL_k = \sum_{i \in T'_k} p_i$ ;

If  $CL_k \geq Q$  Then  $T_k := T_k \cup \{0\}$ ;

If  $CL_k \geq B$  Then 依概率 $P$ 返回节点0,

即 $T_k := T_k \cup \{0\}$ ;

If  $\sum_{i \in M_k(kl)} p_i < B$  Then  $T_k := T_k \cup \{0\}$ , finish := true;

Else

[根据式(13)选择下一节点 $kl$ ;

If  $kl$ 满足连续同宽轧长限制, 且 $CL_k +$

$l_{kl} \leq Q$  Then  $T_k := T_k \cup \{kl\}$ ;

更新  $M_k(kl)$ ;

While ( NOT finish )

End-for

3° 应用局部搜索过程优化各蚂蚁构造的解;

4° For  $k := 1$  To  $m$  Do

依式(12)计算每个蚂蚁构造解的目标函数值;

End-for

如果当前最优解得到改进, 则更新  $T_{gb}$ ;

针对每一弧  $(i, j)$ , 依式(14)更新信息素浓度;

5° If 满足终止条件 Then 输出  $T_{gb}$ ; Else 转步2°.

### 3.2 局部搜索过程(Local search procedure)

为了改进算法的运行效果, 作者在ACO算法中嵌入了两种局部搜索过程——单向插入和2-opt局部搜索过程. 单向插入局部搜索过程就是将未安排的板坯加入到已安排的计划单元中来对解进行改进. 记  $s$  为蚂蚁构造的解,  $M$  为未被访问的节点集合,  $N$  为  $s$  中的节点数(在局部搜索过程中动态更新), 则单向插入搜索过程流程如下:

#### 过程 2

For  $i := 1$  To  $|M|$

For  $j := 1$  To  $N$

[If 将节点  $i$  插入  $s$  中的位置  $j$  满足单元最大轧制长度限制和连续同宽轧制长度限制, 并对解  $s$  有所改进

Then 将节点  $i$  插入  $s$  中的位置  $j$ ;

Return  $s$ .

2-opt局部搜索过程就是对单向插入局部搜索过程改进后的解中的每个轧制单元, 应用求解TSP的2-opt局部搜索算子, 对每个轧制单元内板坯的排列顺序进行再改进.

## 4 实验结果(Experimental results)

为了测试模型与算法的优化效果及时间效率, 以某钢铁企业216个实际板坯为原始数据, 应用本文方法编制热轧生产批量计划. 算法应用Java语言编写, 运行于使用Windows 2000操作系统的Pentium IV(2.6G/512M)PC机上.

### 4.1 对照算法(Comparative algorithm)

实验中, 针对本文模型设计了简单遗传算法<sup>[14]</sup>来求解同一问题作为对比. 算法采用双链表编码方式, 个体  $I = (\pi, l)$  由可重复自然数链表  $\pi = (e_1, e_2, \dots, e_n)$  和非重复自然数链表  $l = (j_1, j_2, \dots, j_n)$  构成. 链表  $\pi$  中第  $i$  个基因  $e_i = 0$  代表板坯  $i$  不被编入轧制单元计划,  $e_i = k, k = 1, 2, \dots, K$  代表板坯  $i$  被编入第  $k$  个轧制单元计划. 链表  $l$  中的基因值为板坯编号, 基因位置对应板坯在轧制单元计划中排

列的先后顺序. 解码过程是依据板坯在链表  $\pi$  中对应的基因值  $k$  和在链表  $l$  中的排列顺序形成若干个轧制单元计划. 如果某个个体通过以上解码过程得到的轧制单元计划满足约束(8)~(10), 称其为可行个体, 否则称其为不可行个体. 定义不可行个体的适值为0, 可行个体的适值根据下式计算:

$$f(I) = f^{\max} - Z(I). \quad (16)$$

其中:  $f^{\max}$  为当前一代种群中所有可行个体解码后得到的轧制批量计划依据式(12)计算的目标函数值的最大值,  $Z(I)$  为由可行个体  $I$  解码后得到的轧制批量计划依据式(12)计算的目标函数值.

遗传算法的交叉和变异算子设计如下: 对于链表  $\pi$  采用一点交叉算子, 变异算子是依变异概率随机改变各基因值. 对于链表  $l$ , 采用求解TSP问题时广泛应用的部分匹配交叉(PMX)算子和2-opt算子. 选择算子采用轮盘赌方式进行.

### 4.2 计算结果(Computational results)

模型参数设置如下:  $Q = 67, B = 60, R = 15, \lambda = 2000$ . ACO算法参数设置方法如下: 在实验初始阶段设置  $w_1 = w_2 = w_3 = 1/3, m = 20, P = 0.01, \tau_0 = 0.02, \max Iter = 100$ , 研究算法参数  $\alpha, \beta, \rho$  对求解效果的影响. 设置  $\alpha = \{2, 3, 4, 5\}, \beta = \{2, 3, 4, 5\}, \rho = \{0.10, 0.15, 0.20, 0.25, 0.30, 0.35\}$ . 在  $4 \times 4 \times 6 = 96$  种参数组合中的每种组合下算法运行10次, 对平均目标函数值进行对比. 结果表明,  $\alpha, \beta, \rho$  3个参数设置对算法的求解效果未发现规律性的影响. 96组结果显示, 算法在  $\alpha = 2, \beta = 5, \rho = 0.3$  的参数设置下获得了最好的求解效果.

保持  $w_1 = w_2 = w_3 = 1/3, \tau_0 = 0.02, \max Iter = 100, \alpha = 2, \beta = 5, \rho = 0.3$  的参数设置, 研究参数  $m, P$  对算法求解效果的影响. 设置  $m = \{5, 10, 15, 20, 25, 30\}, P = \{0.01, 0.1\}$ , 在  $m, P$  的每组取值下算法运行10次, 结果表明: 当  $m = 20, P = 0.01$  时, 算法运行效果最好. 在运行效率方面,  $P = 0.01$  时运行效率较高.

对于局部搜索过程对算法求解效果的影响研究发现: 嵌入局部搜索过程的ACO算法较没有嵌入局部搜索过程的ACO算法在求解效果上只有较小的改善, 但前者收敛速度更快. 因此, 利用嵌入局部搜索过程的ACO算法求解模型时可以通过减少迭代代数来缩短算法的运行时间.

在ACO算法与遗传算法的对比研究中, ACO算法参数设置为  $\alpha = 2, \beta = 5, \rho = 0.3, \tau_0 = 0.02, \max Iter = 100, m = 20, P = 0.01$ . 遗传算法参数设置如下: 种群规模为20, 迭代代数为100, 交叉概率为0.95, 变异概率为0.05. 为了研究决策者对

优化目标的不同偏好对轧制计划质量的影响, 实验中设计了多组目标权系数组合, 两种算法在每组目标权系数组合下运行10次, 取每组权系数组合下10次最好的运行结果进行对比, 结果如表1所示. 表1中第1~3列是决策者给出的6组目标权系数组合, 第4~9列分别给出了ACO算法得到的轧制计

划中包含的轧制单元数、轧制单元编号、各轧制单元包含的板坯数、各轧制单元的轧制长度、各轧制单元因板坯轧制宽度/硬度/厚度跳跃而受到的总惩罚、因未编入轧制计划的板坯而受到的总惩罚. 表1中第10~15列与第4~9列相对应, 是遗传算法求解得到的各统计值.

表 1 两种算法求解效果对比  
Table 1 Comparisonal results of the two algorithms

| 权系数组合 |       |       | ACO求得的计划 |     |       |      |      |        | GA求得的计划 |     |       |      |      |        |
|-------|-------|-------|----------|-----|-------|------|------|--------|---------|-----|-------|------|------|--------|
| $w_1$ | $w_2$ | $w_3$ | units    | No. | slabs | rl   | p1   | p2     | units   | No. | slabs | rl   | p1   | p2     |
| 0.5   | 0.3   | 0.2   | 3        | 1   | 65    | 66.1 | 1803 | 1413.1 | 3       | 1   | 64    | 65.2 | 1853 | 1541.1 |
|       |       |       |          | 2   | 66    | 66.5 | 1870 |        |         | 2   | 65    | 65.2 | 1890 |        |
|       |       |       |          | 3   | 65    | 65.7 | 1857 |        |         | 3   | 65    | 65.5 | 1876 |        |
| 0.5   | 0.2   | 0.3   | 3        | 1   | 66    | 66.3 | 1989 | 1338.5 | 3       | 1   | 64    | 65.2 | 1993 | 1466.5 |
|       |       |       |          | 2   | 65    | 65.7 | 1954 |        |         | 2   | 65    | 65.9 | 1968 |        |
|       |       |       |          | 3   | 65    | 65.9 | 1907 |        |         | 3   | 65    | 65.5 | 1952 |        |
| 0.3   | 0.5   | 0.2   | 3        | 1   | 65    | 65.5 | 1901 | 1315.1 | 3       | 1   | 65    | 65.2 | 1943 | 1379.1 |
|       |       |       |          | 2   | 67    | 66.8 | 1912 |        |         | 2   | 66    | 67.8 | 1957 |        |
|       |       |       |          | 3   | 66    | 66.4 | 1993 |        |         | 3   | 66    | 65.5 | 2005 |        |
| 0.3   | 0.2   | 0.5   | 3        | 1   | 66    | 66.3 | 2087 | 1253.8 | 3       | 1   | 66    | 67.4 | 2102 | 1445.8 |
|       |       |       |          | 2   | 67    | 66.9 | 2012 |        |         | 2   | 65    | 65.6 | 2053 |        |
|       |       |       |          | 3   | 67    | 66.5 | 2096 |        |         | 3   | 66    | 66.3 | 2128 |        |
| 0.2   | 0.5   | 0.3   | 3        | 1   | 66    | 66.1 | 2028 | 1314.6 | 3       | 1   | 65    | 65.3 | 2056 | 1442.6 |
|       |       |       |          | 2   | 66    | 66.4 | 2030 |        |         | 2   | 66    | 66.9 | 2070 |        |
|       |       |       |          | 3   | 67    | 66.5 | 2041 |        |         | 3   | 66    | 65.5 | 2068 |        |
| 0.2   | 0.3   | 0.5   | 3        | 1   | 68    | 66.8 | 2102 | 1244.9 | 3       | 1   | 66    | 65.7 | 2139 | 1436.9 |
|       |       |       |          | 2   | 67    | 66.5 | 2058 |        |         | 2   | 67    | 65.8 | 2091 |        |
|       |       |       |          | 3   | 68    | 67.0 | 2082 |        |         | 3   | 67    | 66.2 | 2102 |        |

由表1可见: 不同的目标权系数组合会引导算法的不同优化偏好, 从而为决策者提供更多的选择. 两种算法比较,ACO算法的求解效果优于遗传算法. 原因在于: ACO算法在蚂蚁构造解的过程中很好地处理了约束(8)~(10). 相比之下, 遗传算法处理约束(8)~(10)时显得过于刚性, 这是由遗传算法的特点决定的.实验过程中也发现: ACO算法在运行效率方面也优于遗传算法, 原因在于: 虽然两种算法每次运行产生的解数量相同, 但由于遗传算法每次迭代要进行选择、交叉和变异等操作, 并且在适值计算时需要判断解的可行性. 因此, 遗传算法较ACO算法具有较高的时间复杂性.

5 结论(Conclusion)

本文结合热轧带钢生产实际和热轧生产工艺约束, 考虑了预选池合同数量的可选择性和计划编制的柔性, 权衡热轧生产率、轧辊利用率和轧辊

成本之间的关系, 建立了轧制生产批量计划多目标优化模型. 基于进化计算的最新技术, 设计了嵌入局部搜索过程的ACO算法对模型进行求解, 并和简单遗传算法进行了对比. 实验结果表明本文的模型和算法是令人满意的. 研究发现, 嵌入局部搜索过程后, 扩大了算法的搜索空间, 算法能够以更快的速度收敛.

参考文献(References):

[1] BALAS E. The prize collecting traveling salesman problem[J]. *Networks*, 1989, 19(6): 621 – 636.  
[2] KOSIBA E D, WRIGHT J R. Discrete event sequence as a traveling salesman problem[J]. *Computers in Industry*, 1992, 19(2): 317 – 327.  
[3] LOPEZ L, CARTER M W, GENDREAU M. The hot strip mill production scheduling problem: A tabu search approach[J]. *European Journal of Operational Research*, 1998, 106(2): 317 – 335.  
[4] 刘士新, 周山长, 宋健海, 等. 基于PCTSP的热轧单元计划模型与算法[J]. *控制理论与应用*, 2006, 23(1): 89 – 92.  
(LIU Shixin, ZHOU Shanchang, SONG Jianhai, et al. PCTSP based

- model and algorithm for hot strip rolling unit planning[J]. *Control Theory & Applications*, 2006, 23(1): 89 – 92.)
- [5] 张涛, 王梦光, 杨建夏. 不确定计划数的轧制批量计划的模型与算法[J]. *系统工程学报*, 2000, 15(1): 54 – 60.  
(ZHANG Tao, WANG Mengguang, YANG Jianxia. The model and algorithm for the hot-milling batch planning with uncertain number[J]. *Chinese J of Systems Engineering*, 2000, 15(1): 54 – 60.)
- [6] COWLING P. *Optimization in Industry: Optimization in Steel hot Rolling*[M]. New York: Wiley, 1995.
- [7] CHEN X, WAN W, XU X. Modeling rolling batch planning as vehicle routing problem with time windows[J]. *Computers & Operations Research*, 1998, 25(12): 1127 – 1136.
- [8] TANG L, LIU J, RONG A, et al. A multiple traveling salesman problem model for hot rolling scheduling in Shanghai Baoshan Iron & Steel Complex[J]. *European J of Operational Research*, 2000, 124(2): 267 – 282.
- [9] 刘士新, 宋健海, 唐加福. 蚁群最优化: 模型、算法及应用综述[J]. *系统工程学报*, 2004, 19(5): 45 – 51.  
(LIU Shixin, SONG Jianhai, TANG Jiafu. Ant colony optimization review: Modelling, algorithms and applications[J]. *Chinese J of Systems Engineering*, 2004, 19(5): 45 – 51.)
- [10] DORIGO M, MANIEZZO V, COLORNI A. The ant system: optimization by a colony of cooperating agents[J]. *IEEE Trans on Systems, Man, and Cybernetics – Part B*, 1996, 26(1): 29 – 41.
- [11] LAPORTE G. The vehicle routing problem: an overview of exact and approximate algorithms[J]. *European J of Operational Research*, 1992, 59(2): 345 – 358.
- [12] BULLNHEIMER B, HARTL R F, STRAUSS C. An improved ant system algorithm for the vehicle routing problem [J]. *Annals of Operations Research*, 1999, 89(1): 319 – 328.
- [13] DORIGO M, CARO G D, GAMBARDELLA L M. Ant algorithms for discrete optimization[J]. *Artificial Life*, 1999, 5(3): 137 – 172.
- [14] GOLDBERG D E. *Genetic Algorithms in Search, Optimization, and Machine Learning*[M]. Massachusetts: Addison-Wesley, 1989.

#### 作者简介:

**刘士新** (1968—), 男, 博士, 教授, 从事制造执行系统、项目管理、最优化方法等研究, E-mail: sxliu@mail.neu.edu.cn;

**宋健海** (1965—), 男, 博士, 高级工程师, 从事钢铁企业ERP、MES及企业信息系统整体解决方案等研究, E-mail: songjianhai@baosight.com;

**周山长** (1976—), 男, 硕士, 从事制造执行系统、最优化方法等研究, E-mail: zhou.shanchang@zte.com.cn .

#### (上接第242页)

- [2] DENG Z L, GAO Y, MAO L, et al. New approach to information fusion steady-state Kalman filtering[J]. *Automatica*, 2005, 41(10): 1695 – 1707.
- [3] 邓自立. 自校正滤波理论及其应用——现代时间序列分析方法[M]. 哈尔滨: 哈尔滨工业大学出版社, 2003.  
(DENG Zili. *Self-tuning Filtering Theory with Applications—Modern Time Series Analysis Method*[M]. Harbin: Harbin Institute of Technology Press, 2003.)
- [4] 邓自立, 马建为, 高媛. 两传感器自校正信息融合Kalman滤波器[J]. *科学技术与工程*, 2003, 3(4): 321 – 324.  
(DENG Zili, MA Jianwei, GAO Yuan. Two-sensor self-tuning information fusion Kalman filter[J]. *Science Technology and Engineering*, 2003, 3(4): 321 – 324.)
- [5] 邓自立. 最优估计理论及其应用——建模、滤波、信息融合估计[M]. 哈尔滨: 哈尔滨工业大学出版社, 2005.  
(DENG Zili. *Optimal Estimation Theory with Applications—Modeling, Filtering, and Information Fusion Estimation*[M]. Harbin: Harbin Institute of Technology Press, 2005.)
- [6] LJUNG L. *System Identification, Theory for the User*[M]. Second Edition. Prentice Hall PTR. Beijing: Tsinghua University Press, 1999.
- [7] 陈传璋. 数学分析[M]. 上海: 上海科学技术出版社, 1962.  
(CHEN Chuanzhang. *Mathematical Analysis*[M]. Shanghai: Shanghai Science Technology Press, 1962.)

#### 作者简介:

**邓自立** (1938—), 男, 教授, 博士生导师, 研究方向为状态估计、多传感器信息融合, E-mail: dzl@hlju.edu.cn;

**李春波** (1980—), 女, 硕士研究生, 研究方向为最优和自校正信息融合Kalman滤波, E-mail: lcbgreens@sohu.com.