

钢铁企业板坯动态分配问题的建模与分散搜索算法求解

吕亚娜, 唐立新, 孟盈, 袁媛

(东北大学 a. 工业工程与物流优化研究所, b. 辽宁省制造系统与物流优化重点实验室, 沈阳 110004)

摘要: 板坯动态分配问题是在一定周期内, 将炼钢-连铸工序动态产出的余材板坯合理分配给期货合同、潜在合同或自拟合同, 使加权费用和最小. 对该问题建立 0-1 整数规划模型, 针对问题的 NP- 难求解性, 设计基于多邻域的分散搜索算法对问题近似求解, 并加入随机策略防止算法陷入局部最优. 分别采用模拟数据和实际数据进行测试, 所提出的算法与商业软件 CPLEX 相比, 可在较短时间内获得近优解, 在解的质量和计算时间方面均优于人工方法.

关键词: 余材板坯; 动态分配; 建模; 分散搜索

中图分类号: TP273

文献标志码: A

Modeling and scatter search algorithm for dynamic slab allocation problem in iron and steel enterprises

LÜ Ya-na, TANG Li-xin, MENG Ying, YUAN Yuan

(a. Institute of Industrial Engineering & Logistics Optimization, b. Liaoning Key Laboratory of Manufacturing System and Logistics, Northeastern University, Shenyang 110004, China. Correspondent: TANG Li-xin, E-mail: lixintang@mail.neu.edu.cn)

Abstract: The dynamic slab allocation problem is to assign the surplus slabs, which are dynamically obtained from steel-making and casting process, to the customer-orders, potential-orders and self-designed-orders in a given period. Therefore, a 0-1 integer programming model is formulated. Due to its NP-hardness, a multiple neighborhoods based scatter search algorithm is proposed to solve the problem approximately. Furthermore, to avoid search process being trapped in local optima, a random perturbation strategy is introduced. Both randomly generated instances and practical instances obtained from a large iron-and-steel enterprise are used to test the performance of the proposed algorithm. Compared with commercial soft named CPLEX, the proposed algorithm can get near-optimal solution in a reasonable CPU time. The experiments on the practical data show that the proposed algorithm outperforms the manual methods on solution quality and computation time.

Keywords: surplus slab; dynamic allocation; modeling; scatter search

0 引言

钢铁企业的生产合同以期货合同为主, 期货合同按用户对企业提出的钢材规格、钢级、成分、硬度等要求进行制定, 是面向需求的, 并具有周期性. 理想的钢铁生产过程中, 炼钢连铸工序产出的板坯应全部用于加工期货合同的产品, 如图 1 实线所示. 但是, 由于企业大批量的生产模式与市场多品种小批量的客户需求模式相矛盾、上游生产异常或生产切换导致不符合客户要求的板坯产出等原因, 导致产生余材板坯. 余材板坯的存在对于钢铁企业生产和运作造成了诸多不利影响, 如增加库存成本、延长生产周期、影响

资金流动等. 因此, 如何合理消化余材板坯, 对于实现企业利润最大化、提高用户满意度、提高生产运作水平起到至关重要的作用. 余材板坯的消化方式主要有两种: 1) 期货合同消化方式, 即将余材板坯分配给当前合适的合同(钢级相同或具有较低钢级要求, 规格相符且有欠量的期货合同); 2) 自拟合同消化方式, 即将余材板坯根据其钢级、规格组成自拟合同进行生产. 自拟合同是生产商根据余材板坯的规格制定的生产合同, 是面向库存的, 如图 1 虚线所示.

实际生产中, 期货合同消化方式是首选, 但是受板坯和期货合同的钢级、规格、合同需求量等因素的

收稿日期: 2013-12-07; 修回日期: 2014-04-15.

基金项目: 国家自然科学基金重点项目(71032004); 国家自然科学基金创新研究群体科学基金项目(71321001); 国家 863 计划项目(2013AA040704).

作者简介: 吕亚娜(1985-), 女, 博士生, 从事钢铁生产计划与调度的研究; 唐立新(1966-), 男, 教授, 博士生导师, 从事流程工业生产计划与调度问题等研究.

制约,如果当前不存在合适的合同,则存在两种选择,等待合适合同的到来(潜在合同),或者制定自拟合同。对于前者,长时间等待将带来库存费用,因此,不能无限制地等待下去,而且当一段时间没有合适合同到来时,可能导致板坯库存越来越高,接近或达到上限警戒值。此时,需要采取自拟合同消化方式快速降低库存,如图2所示。但是,设计自拟合同将带来收益损失,因此,对于余材板坯的3种处理方式,分配给期货合同;分配给潜在合同;制定自拟合同;为动态产出的余材板坯选择合适的处理方式,使得匹配费用、库存费用和收益损失达到平衡是本文所要研究的问题。

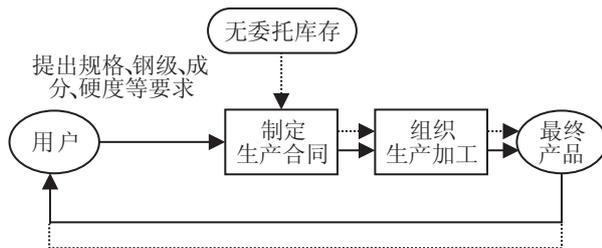


图1 期货合同、自拟合同产生和生产过程示意图

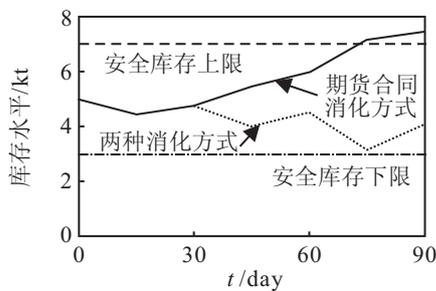


图2 库存变化示意图

国内外已经有许多关于板坯匹配的研究。Kalagnanam等^[1]研究了订单分组的匹配问题,将问题建模为带有颜色约束的多背包问题,并提出基于启发式的网络流以获取近优解。Balakrishnan等^[2]研究了将库存中的板坯分配给合同的问题,考虑订单产品规格的可变性,开发了包含强有效不等式、拉格朗日松弛和启发式方法的综合求解方法。Dawande等^[3]研究了一类板坯设计问题,通过开发启发式求解该问题,与Kalagnanam等^[1]所研究的匹配问题相似,但是其关注的是为合同设计板坯进行生产。Forrest等^[4]针对带有颜色约束的多背包问题提出列生成算法。Yanagisawa^[5]研究了材料分配问题,并提出局域搜索算法,包含丰富的邻域(如链式发射方法)。Tang等^[6]研究了包含所有板坯的全委托匹配问题,改进了板坯的利用率和客户满意度,并采用禁忌搜索算法进行求解。Song^[7]采用列生成算法对板坯可分的剩余板坯匹配问题进行了研究。田志波等^[8]研究了板坯不可分的板坯合同匹配问题,采用基于合成邻域的蚁群算

法求解,在实际中得到了应用,但是该方法考虑期货合同消化方式,未考虑板坯的动态到达、自拟合同消化方式和库存水平约束。Huang等^[9]和Zhang等^[10]集成了板坯匹配和生产计划问题,分析了合同、库存和生产能力之间的关系,决策哪些合同需要用库存交付,哪些合同需要组织生产并安排生产日期,分别采用0-1线性规划、改进遗传算法和混合PSO/TS算法进行求解,但是仅针对用户合同,并不考虑自拟合同。

以上研究主要针对库存板坯和已有合同的静态问题,没有考虑潜在合同和自拟合同,且没有考虑余材板坯的动态到达。本文所研究的板坯动态分配问题是在给定的展望期内,为每个动态到达的板坯确定合理的处理方式和处理时间。对于该问题,在满足钢级质量、规格等材料属性的要求,满足合同完整性、交货期和安全库存水平等实际约束的条件下,以最小化板坯-合同匹配费用、超量供货的折扣费用、合同未完成的惩罚费用、自拟合同的启动费用、库存费用为目标,建立了0-1整数规划模型。为了加速问题求解,基于问题的结构特征,松弛原问题的库存水平约束,按钢级将原问题分解为若干子问题,并设计基于多邻域的分散搜索算法求解子问题。最后,通过贪婪启发式修复子问题得到松弛解,进而得到原问题的近优解。

1 问题及其数学模型

1.1 问题描述

板坯动态分配问题首先需要决策余材板坯的处理方式,然后确定板坯的具体合同和开始处理时间。其优化目标主要考虑板坯-合同匹配费用、期货合同超欠量惩罚、自拟合同启动费用和库存费用。板坯-合同匹配费用由余材板坯与不同期货合同的匹配度决定,匹配度越低,板坯与合同间的钢级和规格差异越小,匹配质量越好;期货合同超欠量惩罚是期货合同超量的费用折扣和合同未完成的惩罚;自拟合同启动费用是指构造自拟合同的生产成本,与自拟合同个数相关;库存费用是由板坯实际在库时间决定的。

该问题主要涉及5类约束:1)唯一性约束,一块板坯最多只能分配给一个合同;2)期货合同处理时间约束,生产要求分配给同一个期货合同的板坯处理时间相同;3)合同完整性约束,在实际生产中,生产管理部门要求合同余材量不超过分配给该合同的最小板坯重量;4)库存水平约束,在每个时间段内剩余库存需满足库存水平上下限要求;5)聚类约束,组成同一自拟合同的板坯其钢级和处理时间必须相同,且必须有重叠的轧制宽度范围。

1.2 符号定义

设 $I = \{1, 2, \dots, N\}$ 为所有板坯的集合; $J = \{1,$

$2, \dots, L$ 为期货合同的集合; $\Omega = \{0, 1, \dots, \sigma\}$ 为计划展望期离散化后时间点的集合. 对于每个板坯 $i \in I$, w_i 和 g_i 为板坯 i 的重量和钢级. 根据轧机的侧压量要求, 板坯 i 最终轧制成品的宽度在一定范围内, 定义为 $[d_i^1, d_i^2]$, 即可配合同宽度的范围. 由于板坯 i 是动态到达的, 令 $a_{it} = 1$ 表示板坯 i 在时刻 t 到达, 对于 $a_{i0} = 1$ 的板坯, 初始在库时间 $\sigma_i \geq 0$, 对于其他板坯, $\sigma_i = 0$. 定义 σ_i^0 为板坯 i 的到达时间, 存在 $\sigma_i^0 = \sum_{t=0}^{\sigma} ta_{it}$, 得到板坯 i 在时刻 t 的实际在库时间 $\tau_{it} = t - \sigma_i^0 + \sigma_i$. σ_i^1 为板坯 i 等待潜在合同的最大允许时间, σ_i^2 为板坯 i 允许组成自拟合同的最小在库时间, 其中 $\sigma_i^2 \geq \sigma_i^0$. 对于每个期货合同 $j \in J$, o_j 为期货合同 j 的欠量, δ_j^0 为期货合同 j 的最晚处理时间. 定义板坯库在 t 时刻的存储能力范围为 $[U_t^L, U_t^H]$. 定义如下模型参数: F_1 为匹配费用的权重, F_2 为分配给期货合同的超欠量费用权重, F_3 为组成自拟合同的启动费用权重, F_4 为板坯库存费用权重; C_{ij}^1 为板坯 i 分配给期货合同 j 的匹配费用, C_{it}^2 为板坯 i 在时刻 t 分配给潜在合同的匹配费用, C_{it}^3 为板坯 i 组成自拟合同的匹配费用; q_1 和 q_2 分别为期货合同超量和欠量的单位(吨)惩罚费用; p 为自拟合同的启动费用; h 为单位时间库存费用. 定义 M 为一个很大的正数.

问题的决策变量为 x_{ijt} , y_{it} , z_{ikt} , 表示板坯的具体处理方式和处理时间, 当板坯 i 在时刻 t 分配给期货合同 j 时, $x_{ijt} = 1$, 否则为 0; 当板坯 i 在时刻 t 分配给潜在合同时, $y_{it} = 1$, 否则为 0; 当板坯 i 在时刻 t 组成自拟合同且聚类在第 k 个板坯时, $z_{ikt} = 1$, 否则为 0.

1.3 数学模型

基于上面定义参数和决策变量, 结合板坯分配过程中考虑的目标及约束, 对钢铁企业实际存在的板坯动态分配问题建立 0-1 整数规划模型

$$\begin{aligned} \min Z = & F_1 \sum_{i=1}^N \left(\sum_{j=1}^L \sum_{t=\sigma_i^0}^{\delta_j^0} C_{ij}^1 x_{ijt} + \sum_{t=\sigma_i^0}^{\sigma_i^1} C_{it}^2 y_{it} + \sum_{k=1}^{\sigma} \sum_{t=\sigma_i^2}^{\sigma} C_{it}^3 z_{ikt} \right) + F_2 \sum_{j=1}^L (q_1 \text{Over}_j + q_2 \text{Lack}_j) + \\ & F_3 \sum_{t=\sigma_i^2}^{\sigma} \sum_{k=1}^N p z_{kkt} + F_4 \left(\sum_{i=1}^N \sum_{j=1}^L \sum_{t=\sigma_i^0}^{\delta_j^0} h \tau_{it} w_i x_{ijt} + \sum_{i=1}^N \sum_{t=\sigma_i^0}^{\sigma_i^1} h \tau_{it} w_i y_{it} + \sum_{i=1}^N \sum_{k=1}^N \sum_{t=\sigma_i^2}^{\sigma} h \tau_{it} w_i z_{ikt} \right). \quad (1) \end{aligned}$$

s. t.

$$\sum_{j=1}^L \sum_{t=\sigma_i^0}^{\delta_j^0} x_{ijt} + \sum_{t=\sigma_i^0}^{\sigma_i^1} y_{it} + \sum_{k=1}^N \sum_{t=\sigma_i^2}^{\sigma} z_{ikt} = 1, \quad i \in I; \quad (2)$$

$$M \left(2 - \sum_{t=\sigma_i^0}^{\delta_j^0} x_{ijt} - \sum_{t=\sigma_{i'}}^{\delta_{j'}} x_{i'jt} \right) \geq \sum_{t=\sigma_i^0}^{\delta_j^0} t x_{ijt} - \sum_{t=\sigma_{i'}}^{\delta_{j'}} t x_{i'jt}, \quad i, i' \in I, j \in J; \quad (3)$$

$$\sum_{i=1}^N \sum_{t=\sigma_i^0}^{\delta_j^0} w_i x_{ijt} - o_j < m w_j, \quad j \in J; \quad (4)$$

$$U_t^L \leq U_t \leq U_t^H, \quad t \in \Omega; \quad (5)$$

$$z_{ikt} \leq z_{kkt}, \quad i, k \in I, t \in \Omega; \quad (6)$$

$$d_i^1 z_{ikt} \leq d_{i'}^2 + (1 - z_{i'kt}) M, \quad i, i', k \in I, t \in \Omega; \quad (7)$$

$$g_i z_{ikt} \leq g_{i'} + (1 - z_{i'kt}) M, \quad i, i', k \in I, t \in \Omega; \quad (8)$$

$$x_{ijt}, y_{it}, z_{ikt} \in \{0, 1\}, \quad i, k \in I, j \in J, t \in \Omega. \quad (9)$$

其中

$$\text{Over}_j = \max \left\{ \left(\sum_{i=1}^N \sum_{t=\sigma_i^0}^{\delta_j^0} w_i x_{ijt} - o_j \right), 0 \right\},$$

$$\text{Lack}_j = \max \left\{ \left(o_j - \sum_{i=1}^N \sum_{t=\sigma_i^0}^{\delta_j^0} w_i x_{ijt} \right), 0 \right\}, \quad (10)$$

$$m w_j = \min \{ w_i | x_{ijt} = 1, i \in I, t \in [\sigma_i^0, \delta_j^0] \}, \quad (11)$$

$$U_t = \sum_{i=1}^N \sum_{t'=0}^t w_i a_{it'} - \sum_{i=1}^N \sum_{j=1}^L \sum_{t'=\sigma_i^0}^t w_i x_{ijt'} - \sum_{i=1}^N \sum_{t'=\sigma_i^0}^t w_i y_{it'} - \sum_{i=1}^N \sum_{k=1}^N \sum_{t'=\sigma_i^2}^t w_i z_{ikt'}. \quad (12)$$

目标函数(1)中, 第1项为板坯匹配费用, 第2项为期货合同超欠量惩罚, 第3项为自拟合同启动费用, 第4项为板坯库存惩罚; 约束(2)为唯一性约束; 约束(3)为期货合同处理时间约束; 约束(4)为期货合同完整性约束; 约束(5)为库存水平约束; 约束(6)~(8)为自拟合同聚类约束; 约束(9)为决策变量的取值范围.

若只考虑期货合同, 将板坯视为物品, 将合同视为背包, 板坯与合同的匹配费用作为物品装包的负价值, 以期货合同的欠量作为背包的容量, 则该问题可以看作是一类多背包问题, 潜在合同和自拟合同可以看作是没有容量限制的背包. 由于问题的决策变量只能取 0 或 1, 该问题可归结为 0-1 多背包问题. 经众多学者研究并证明, 0-1 多背包问题为 NP-hard 问题^[11], 因此, 板坯动态分配问题为 NP-hard 问题.

2 问题求解

由于钢级不同且不能以优充次的板坯与合同不存在匹配关系,并且在组成自拟合同时,只有相同钢级的板坯才能分配在同一自拟合同中,本文将余材板坯的库存水平约束松弛,然后根据板坯与合同的钢级将问题分解为若干个子问题.将问题分解可以大大降低板坯与合同的匹配规模,提高求解速度.对于分解后的子问题,本文采用分散搜索算法进行求解,根据问题特征构造3种邻域结构,并加入随机策略防止算法过早陷入局部最优.由于将库存水平约束松弛后,简单地将各个子问题的解进行整合后得到的解不一定是原问题的可行解,当得到的解不可行时,本文构造一个贪婪启发式算法对解进行修复,以满足库存水平约束.

2.1 分散搜索算法基本思想

分散搜索(SS)算法^[12-13]是进化计算领域求解组合优化问题的一类优化算法. SS是由Glover^[12]为解决整数规划问题而提出,目前已成功应用于许多组合优化问题,如求解聚类^[14]、项目调度^[15]、车辆路径^[16]和车间任务调度^[17]等.分散搜索算法是基于种群的进化算法,该算法首先创建初始种群,并从该种群中选择质量好的和分散性强的解组成参考集.根据参考集中的新解构造子集,组合子集产生新解,改进解后更新参考集.其基本思想是,通过对参考集中解的质量和分散性的控制,尽量产生分散于整个解空间的局部搜索算法的初始解,从而使得搜索的解空间范围大,以获得更好的搜索结果.组合解和更新参考集的方式使SS不同于遗传算法等其他进化算法.

2.2 分散搜索算法求解板坯动态分配问题

本文采用分散搜索算法对分解后的子问题进行求解.在求解过程中根据问题的特点,采用三基因结构链表示一个匹配方案,如图3所示.每个板坯对应一个三基因结构.图3中:1表示板坯所采用的处理方式;当板坯分配给期货合同时,2表示具体合同;当板坯分配给潜在合同时,2没有意义;当板坯组成自拟合同时,2表示聚类号(即自拟合同号);3表示板坯的处理时间.由于期货合同的处理时间与具体的合同有关,当具体合同确定后处理时间即可得知,同样,当板坯组成自拟合同的处理时间确定后自拟合同的聚类号也可计算得到.定义匹配节点 i, j, k_m ,当 $j = 1$ 时,表示将板坯 i 分配给第 k_m 个期货合同;当 $j = 2$ 时,表示将板坯 i 在 k_m 时刻分配给潜在合同;当 $j = 3$ 时,表示将板坯 i 在 k_m 时刻组成自拟合同.根据匹配条件,若匹配节点可行,则该节点称为可配节点,否则为

不可配节点.将可配节点按照匹配费用从小到大排序.

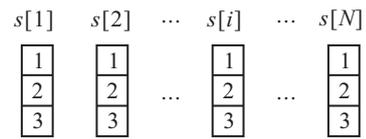


图3 解的三基因结构

2.2.1 构造初始种群

设初始种群的大小为 P_{size} .为了产生好的初始种群,本文设计启发式 H 以获得 P_1 个质量较好的解,同时通过随机策略产生 P_2 个分散性好的解,这样可以同时保证初始种群中解的质量和分散性.

启发式 H 定义如下:分类依次遍历期货合同、潜在合同和自拟合同的可配节点;若板坯未配,则选择该可配节点;若板坯已配,则继续下一个可配节点,直到所有可配节点遍历完成后,执行聚类启发式 J .对于所有分配给自拟合同的板坯,按照钢级、处理时间、轧制宽度进行聚类.

调整启发式 H 中遍历3类可配节点的顺序,可以产生不同的初始解,选取 P_1 个质量好的解放入初始种群.另外, P_2 个分散性好的解的产生方法如下:循环所有板坯,为每一个板坯随机选择可配节点,直到所有的板坯循环完成,执行聚类启发式 J .

自拟合同的设计与板坯钢级、处理时间和轧制宽度范围有关,因此,按照钢级和处理时间对板坯分组,每组板坯按照轧制宽度范围进行聚类.聚类约束为宽度范围有重叠,目标为聚类个数最小,针对每组板坯提出多项式时间算法.

聚类启发式 J 定义如下(设每个分类板坯的个数为 h).

Step 1: 初始化.设板坯 i 的聚类号 $cluster[i] = 0$, $i = 1, 2, \dots, h$.构造宽度值序列 $wseq$,用来存储所有板坯的轧制宽度上限和下限值.将 $wseq$ 从小到大进行排序,若宽度值相同则下限值在前,上限值在后. $wseq[l]$ 表示第 l 个宽度值,令 $s[l]$ 表示 $wseq[l]$ 对应的板坯序号.设 $l = 1$.

Step 2:若 $l \leq 2h$,则转至Step 3,否则,结束聚类.

Step 3:如果 $cluster[s[l]]$ 不为0, $l++$,则转至Step 2.否则,寻找 $wseq$ 中序号 l 后面的第1个序号 k ,满足 $cluster[s[k]] = 0$,且 $wseq[k]$ 为宽度上限值.令 $cluster[m] = s[l]$,其中 $m \in \{s[l], s[l+1], \dots, s[k]\}$,且满足 $cluster[m] = 0$.令 $l = k + 1$,转至Step 2.

图4为4个板坯 $s_1 \sim s_4$ 的聚类过程.按照上述启发式,最后产生两个聚类, s_1 和 s_2 组成聚类1, s_3 和 s_4 组成聚类2.

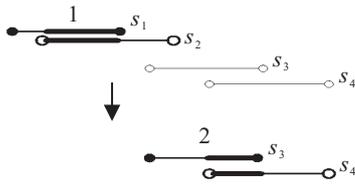


图 4 聚类启发式示意图 (4 个板坯)

2.2.2 创建参考集和产生子集

分散搜索算法中, 参考集中解的选取直接影响算法的性能, 因此在创建参考集时, 需要同时兼顾参考集中解的质量和分散性. 解之间的距离是判断分散性的关键, 针对板坯动态分配问题, 两个解之间的距离定义为分配方案不相同的板坯的个数. 对于解 S 和 S' , 其距离可以通过下式计算:

$$D(S, S') = \sum_{i=1}^N \left(\sum_{j=1}^L \sum_{t=\sigma_j^0}^{\delta_j^0} |x_{ijt} - x'_{ijt}| + \sum_{t=\sigma_i^0}^{\sigma_i^1} |y_{it} - y'_{it}| + \sum_{k=1}^N \sum_{t=\sigma_k^2}^{\sigma} |z_{ikt} - z'_{ikt}| \right).$$

定义一个解 S 与一个集合的距离为 S 距离集合中所有解的最小距离. 根据上述距离定义, 参考集 RefSet 构造如下:

- 1) 将初始种群中的解按目标函数从小到大排序, 选择前 b_1 个质量好的解放入参考集;
- 2) 循环 b_2 次, 每次选择距离 RefSet 最远的解放入参考集, 若最远的解不止一个, 则随机选择其中一个放入参考集.

通过上述步骤可以得到一个有 b 个解的参考集, 该参考集中包含 b_1 个质量好的解, b_2 个分散性强的解. 参考集创建后, 将参考集中的解两两组合, 构成二元子集.

2.2.3 组合子集和更新参考集

构成二元子集后, 组合子集中的解产生新解, 新解继承子集中解的部分特性, 同时也具有新的特性. 组合方法如下:

- 1) 若板坯的匹配节点完全相同, 则在新解中仍然采用该节点;
- 2) 若分配的合同类型相同, 但是具体的合同号或处理时间不同, 则选择匹配费用较小的匹配节点;
- 3) 若分配的合同类型也不同, 则随机选择其中一个匹配节点.

由于组合时没有考虑期货合同完整性约束, 可能存在合同超量的情况, 还需要对组合后的新解进行修复. 修复过程如下: 找出所有超量合同上的板坯, 依次处理每个超量合同, 将分配给同一期货合同的板坯按

照重量从小到大排序, 从第 1 个板坯开始, 若移除该板坯后合同仍超量, 则将该板坯的分配关系解除, 直至移除该板坯后合同不超量; 然后, 对解除分配关系的板坯在其潜在合同和自拟合同可配节点中, 选择一个匹配费用较小的节点, 并执行聚类启发式 J 进行聚类.

参考集的更新采用静态更新方式, 即当所有子集组合产生新解后, 将比参考集中解的质量好的新解按照目标函数从小到大的顺序放入参考集, 并移除原有参考集中最差的解. 此外, 需要对更新信息进行记录, 防止重复组合子集.

2.2.4 局部搜索算法

局部搜索算法的核心是构造邻域. 针对问题的特点, 本文设计并构造移动邻域、交换邻域和环交换邻域 3 种邻域结构. 移动邻域分为组内移动和组间移动, 如图 5(a) 和图 5(b) 所示. 交换邻域分为一对一交换、二对一交换和环交换, 如图 5(c)~图 5(e) 所示.

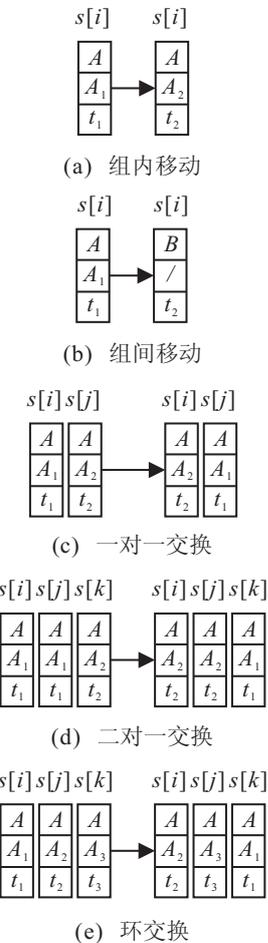


图 5 板坯动态分配问题的邻域结构

组内移动表示板坯所属合同类不变, 仅调整所分配的具体合同或时间, 如板坯 $s[i]$ 在合同类 A 中由在时刻 t_1 分配给期货合同 A_1 调整为在时刻 t_2 分配给期货合同 A_2 . 组间移动表示变换板坯所属的合同类, 并在变换后的合同类中选择一个可配节点, 图 5(b) 表

示板坯 $s[i]$ 由分配给合同类 A 变为分配给合同类 B . 一对一交换表示两个板坯交换匹配节点, 二对一交换表示两个板坯和一个板坯交换匹配节点, 主要针对分配给期货合同类的板坯, 图 5(d) 中板坯 $s[i]$ 和 $s[j]$ 在时刻 t_1 分配给合同 A_1 , 板坯 $s[k]$ 在时刻 t_2 分配给合同 A_2 , 交换后则变为板坯 $s[i]$ 和 $s[j]$ 在时刻 t_2 分配给合同 A_2 , 板坯 $s[k]$ 在时刻 t_1 分配给合同 A_1 . 环交换是指 3 个或 3 个以上板坯之间的循环交换, 本文只采用 3 个板坯的环交换, 图 5(e) 表示交换后板坯 $s[i]$ 为板坯 $s[j]$ 原来的可配节点, 以此类推.

基于上述 3 种邻域结构的局部搜索算法如下: 对解依次执行组间移动、组内移动、一对一交换、二对一交换和环交换, 若目标函数有改进, 则执行该邻域, 重复此步骤, 直到没有改进或局域搜索最大循环次数 CLS 达到设定值为止.

2.2.5 重建初始种群

为了防止算法陷入局部最优, 需要对算法进行扰动, 这里采用重建初始种群方法重新调整初始搜索空间. 重建初始种群时, 保留历史最好解, 具体实施方法如下:

- 1) 清空初始解池;
- 2) 将历史最好解放入解池中;
- 3) 采用随机策略产生 $P_{\text{size}} - 1$ 个解放入解池中.

2.2.6 修复不可行解

分散搜索算法用于求解将库存水平约束松弛并按钢级分组后的动态板坯分配子问题, 因此将所有子问题的解进行简单整合后的解不一定满足库存水平约束, 基于目标函数增量最小的思想, 设计贪婪启发式算法对松弛解进行调整.

计算松弛解所有时间段的库存下限偏差 U_t^{\min} 和上限偏差 U_t^{\max} , 由式 (12) 可得

$$U_t^{\min} = U_t - U_t^L, U_t^{\max} = U_t^H - U_t.$$

设集合 A 和 B 分别表示 $U_t^{\min} < 0$ 和 $U_t^{\max} < 0$ 的时间集合, 若 A 和 B 都为空, 则松弛解可行, 否则需要对不可行解进行修复.

贪婪启发式 T 步骤如下.

Step 1: 若 A 和 B 都不为空, 则将处理时间属于 A 的所有板坯依次从当前可配节点移动到处理时间属于 B 的可配节点, 计算目标函数增量.

Step 2: 若 A 不为空, B 为空, 则将处理时间属于 A 的所有板坯依次从当前可配节点移动到处理时间不属于 A 的可配节点, 计算目标函数增量.

Step 3: 若 A 为空, B 不为空, 则将处理时间不属于 B 的所有板坯依次从当前可配节点移动到处理时间属于 B 的可配节点, 计算目标函数增量.

Step 4: 执行增量最小且满足之前处理时间库存约束的移动, 更新相应的 U_t^{\min} 、 U_t^{\max} 、 A 和 B .

Step 5: 若 A 或 B 不为空, 则转至 **Step 1**, 否则松弛解可行; 若有涉及自拟合同的板坯分配方式发生改变, 则执行聚类启发式 J , 修复结束.

2.3 算法整体框架

本文采用最好解连续未改进次数 C_{Best} 和最大运行时间为停止准则. 板坯动态分配问题的算法描述如下.

Step 1: 将板坯合同数据按照钢级进行分组.

Step 2: 对每组数据执行分散搜索算法.

Step 2.1: 初始化, 令 $C_{\text{Best}} = 0$;

Step 2.2: 产生初始种群;

Step 2.3: 构造初始参考集 RefSet;

Step 2.4: 产生二元子集, 并组合子集产生新解, 用局部搜索方法改进新解, 将改进解放入候选集;

Step 2.5: 更新参考集, 若最好解被更新, 则 $C_{\text{Best}} = 0$; 否则 $C_{\text{Best}}++$, 若参考集有更新, 则执行 **Step 2.4**, 否则执行 **Step 2.6**;

Step 2.6: 若满足停止准则, 则执行 **Step 2.7**, 否则重建初始种群, 返回 **Step 2.3**;

Step 2.7: 若所有钢级分组数据计算完成, 则执行 **Step 3**, 否则继续下一分组数据, 执行 **Step 2.1**.

Step 3: 执行贪婪启发式 T 修复松弛解, 得到整个问题的可行解.

3 实验分析

算法在 VC++ 6.0 环境下采用 C++ 语言编程实现, 并在配置为双核、Q9550 CPU 处理器、2.83 GHz 主频和 3.25 GB 内存的计算机上进行仿真实验. 所有测试数据均采用标准求解软件 CPLEX 进行求解.

采用国内某大型钢厂实际的板坯匹配数据和随机生成的模拟数据对算法进行测试, 实际数据取 3 个月的生产数据, 根据生产实际, 以 15 天为一个时间单位, 将展望周期分为 7 个处理时间段, 即 $[0, 15)$, $[15, 30)$, \dots , $[90, \infty)$. 模拟数据板坯规模为板坯个数 $N \in \{10, 20, \dots, 100\}$, 合同个数 $L = N \times \rho\%$. 其中: $\rho \in \{20, 40, 60\}$, 计划展望期 $\sigma = 6$. 每个规模产生 10 组数据, 对每组数据进行 10 次测试求平均值. 模拟数据根据实际随机产生, 设每组数据的钢级相同, 其他参数随机产生, 范围为: $\sigma_i^0 \in [0, 6]$, 当 $\sigma_i^0 = 0$ 时, $\sigma_i \in [0, 6]$, 当 $\sigma_i^0 > 0$ 时, $\sigma_i = 0$; $\sigma_i^1 \in [0, 6]$; $\sigma_i^2 \in [\sigma_i^0, 6]$; $w_i \in [10, 35]$; $d_i^1 \in [900, 1800]$; $d_i^2 = d_i^1 + \Delta$, $\Delta \in [0, 200]$; $\delta_j^0 \in [0, 6]$; $o_j \in [10, 100]$; $U_t^L \in \left[0, 0.3 \sum_{i=1}^N w_i\right]$, $U_t^H \in$

$\left[0.7 \sum_{i=1}^N w_i, \sum_{i=1}^N w_i\right]$; $C_{ij}^1, C_{it}^2 \in [0, 40]$, $C_{it}^3 \in [0, 20]$;
 $q_1 = 16, q_2 = 24; p = 0.1; h = 0.075$. 设权重系数相等, 均为 0.25.

利用分散搜索算法求解板坯匹配问题时, 算法参数根据多次实验结果进行设置, 具体如下:

- 1) 初始种群的大小 $P_{size} = 20, P_1 = 3$;
- 2) 参考集的大小 $b = 5$, 质量好的解的数目 $b_1 = 3$, 分散性好的解的数目 $b_2 = 2$;
- 3) 局域搜索最大循环次数 CLS 为 3;
- 4) 最好解连续未改进最大次数 C_{Best} 为 10;
- 5) 最大运行时间为 600 s.

表 1 为中小规模数据测试结果. 表 1 中: AVG 和 MAX 分别表示算法平均值和最大值与 CPLEX 求出的最优解的比值, T 为平均运行时间的比值. 由于算法求得的最好解与 CPLEX 求出的最优解相同, 即平均最好解比值为 1, 因此表中不再赘述. 由表 1 可见, 对于中小规模算例, 分散搜索算法获得的解与最优解的平均比值仅为 1.001, 平均最劣解比值为 1.004. 其中, 所提出算法在求解 60% N 和 40% N 合同规模算例时, 求解时间分别为 CPLEX 的 25.2% 和 29.7%.

表 1 CPLEX和SS求解中小规模模拟数据测试结果对比

| N | 60% | | | 40% | | | 20% | | |
|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | AVG | MAX | T | AVG | MAX | T | AVG | MAX | T |
| 10 | 1.000 | 1.001 | 0.299 | 1.000 | 1.000 | 0.413 | 1.000 | 1.000 | 0.512 |
| 20 | 1.000 | 1.001 | 0.416 | 1.000 | 1.001 | 0.622 | 1.000 | 1.000 | 2.288 |
| 30 | 1.001 | 1.002 | 0.281 | 1.001 | 1.002 | 0.426 | 1.000 | 1.000 | 2.319 |
| 40 | 1.001 | 1.002 | 0.209 | 1.003 | 1.012 | 0.093 | 1.002 | 1.005 | 2.373 |
| 50 | 1.003 | 1.006 | 0.159 | 1.004 | 1.010 | 0.129 | 1.001 | 1.003 | 1.174 |
| 60 | 1.002 | 1.006 | 0.146 | 1.005 | 1.013 | 0.099 | 1.002 | 1.004 | 1.260 |
| Avg. | 1.001 | 1.003 | 0.252 | 1.002 | 1.006 | 0.297 | 1.001 | 1.002 | 1.654 |

在同等测试条件下, CPLEX 无法求解的大规模算例, 所提出的分散搜索算法仍然可以在较短的时间内获得问题的满意解. 表 2 所示的大规模数据测试结果中, AVG 和 MAX 分别表示算法平均值和最大值与算法最小值的比值, TIME 为算法平均运行时间. 由表 2 可见, 大规模数据中, 算法获得的解的平均值比值与最大值比值为 1.006 和 1.012, 其中 60% N 和 40% N 合同规模算例的求解时间均控制在 60 s 以内.

表 2 SS算法求解大规模模拟数据的测试结果

| N | 60% | | | 40% | | | 20% | | |
|------|-------|-------|--------|-------|-------|--------|-------|-------|--------|
| | AVG | MAX | TIME/s | AVG | MAX | TIME/s | AVG | MAX | TIME/s |
| 70 | 1.002 | 1.004 | 9.375 | 1.009 | 1.021 | 15.39 | 1.002 | 1.006 | 110.8 |
| 80 | 1.002 | 1.005 | 12.54 | 1.011 | 1.023 | 31.71 | 1.002 | 1.005 | 207.1 |
| 90 | 1.002 | 1.004 | 22.86 | 1.013 | 1.026 | 55.17 | 1.004 | 1.009 | 330.9 |
| 100 | 1.001 | 1.002 | 31.35 | 1.013 | 1.027 | 51.75 | 1.007 | 1.013 | 500.7 |
| Avg. | 1.002 | 1.004 | 19.03 | 1.012 | 1.024 | 38.51 | 1.004 | 1.008 | 287.38 |

表 3 为不同 SS 算法求解模拟数据的比较结果. 表 3 中: AVG1、AVG2 和 AVG3 分别表示传统分散搜索、仅加入重建初始种群方法的分散搜索、仅加入局域搜索的分散搜索与最终分散搜索算法的平均目标比值, Avg. 为目标比值的平均值, Avg.T 为平均时间比值. 由表 3 可见: 加入局域搜索的分散搜索算法较为费时, 但是解的质量较好; 仅采用重建初始种群方法的算法求解速度很快, 但质量稍差; 传统分散搜索算法求解时间最短, 其求解质量也是最差的; 最终分散搜索算法的求解时间在可接受范围内, 求解质量也最好. 实际应用中, 可以根据不同的需要调整不同策略使用的比重以达到求解目的.

表 3 不同SS算法求解模拟数据的比较结果

| N | 60% | | | 40% | | | 20% | | |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | AVG1 | AVG2 | AVG3 | AVG1 | AVG2 | AVG3 | AVG1 | AVG2 | AVG3 |
| 10 | 1.212 | 1.178 | 1.001 | 1.266 | 1.231 | 1.000 | 1.586 | 1.163 | 1.000 |
| 30 | 1.306 | 1.300 | 1.001 | 1.531 | 1.466 | 1.001 | 2.044 | 1.438 | 1.000 |
| 50 | 1.336 | 1.323 | 1.001 | 1.916 | 1.832 | 1.004 | 2.276 | 1.671 | 1.001 |
| 80 | 1.443 | 1.438 | 1.002 | 2.469 | 2.395 | 1.009 | 2.572 | 1.839 | 1.002 |
| 100 | 1.314 | 1.314 | 1.001 | 2.396 | 2.386 | 1.004 | 2.811 | 1.926 | 1.001 |
| Avg. | 1.322 | 1.311 | 1.001 | 1.916 | 1.862 | 1.004 | 2.258 | 1.607 | 1.001 |
| Avg. T | 0.002 | 0.011 | 0.420 | 0.001 | 0.005 | 0.451 | 0.001 | 0.006 | 0.823 |

由于实际数据规模较大, CPLEX 无法在允许的时间和硬件条件下求解, 采用分散搜索算法分别对 15 天、30 天、60 天、90 天的实际数据进行测试, 并与人工方法结果相比较, 如表 4 所示. 表 4 中: 第 1 列和第 2 列表示数据规模; 第 3 列~第 6 列表示各指标算法与人工结果的偏差((算法 - 人工)/人工); 第 7 列为算法运算时间. 由表 4 可见, 算法在多数指标上相对于人工结果有较大下降, 由于自拟合同启动费用设置较小, 该指标的改进不十分明显. 实际应用中, 可以根据不同的需求调整权重, 从而得到满意的结果. 算法求解时间也在可接受范围内, 满足了实际生产方面的要求.

表 4 实际数据测试结果

| 天数 | $N \times L$ | 匹配费用 | 超欠量费用 | 自拟合同启动费用 | 库存费用 | 时间/s |
|----|----------------|--------|--------|----------|--------|-------|
| 15 | 10 410 × 4 148 | -0.432 | -0.471 | -0.062 | -0.536 | 50.7 |
| 30 | 11 021 × 4 558 | -0.435 | -0.491 | -0.015 | -0.531 | 54.9 |
| 60 | 17 930 × 8 449 | -0.413 | -0.214 | 0.029 | -0.525 | 111.3 |
| 90 | 20 290 × 9 992 | -0.416 | -0.182 | -0.042 | -0.512 | 84.7 |

4 结 论

本文研究了钢铁企业生产过程中普遍存在的板坯动态分配问题, 目标是平衡板坯采用不同处理方式时的生产和库存费用, 对板坯进行生产定位. 本文根据实际匹配过程中的约束及目标, 建立了 0-1 整数规划模型, 并设计了基于多邻域的分散搜索算法进行求

解. 采用基于3种邻域结构的局域搜索策略和随机策略增加算法寻优能力, 并防止算法陷入局部最优. 使用实际数据和模拟数据对算法进行测试, 同时采用CPLEX对模拟数据进行求解. 当CPLEX可以获得最优解时, 所提出的算法也可以获得最优解, 并且在较大规模的算例中求解速度更快. 对于CPLEX无法求解的算例, 所提出的算法依旧可以在合理的时间内得出近似解. 本文采用的分散搜索算法的求解质量好于单纯使用局域搜索或重建初始种群方法的算法. 对于实际算例, 所提出的算法与人工方法相比, 可以有效降低板坯-合同匹配费用、库存费用、合同余材量等经济技术指标, 具有显著的实际应用价值.

参考文献(References)

- [1] Kalagnanam J R, Dawande M W, Trumbo M, et al. The surplus inventory matching problem in the process industry[J]. *Operations Research*, 2000, 48(4): 505-516.
- [2] Balakrishnan A, Genunes J. Production planning with flexible product specifications: An application to specialty steel manufacturing[J]. *Operations Research*, 2003, 51(1): 94-112.
- [3] Dawande M, Kalagnanam J, Lee H S, et al. The slab-design problem in the steel industry[J]. *Interfaces*, 2004, 34(3): 215-225.
- [4] Forrest J J H, Kalagnanam J, Ladanyi L. A column-generation approach to the multiple knapsack problem with color constraints[J]. *Informs J on Computing*, 2006, 18(1): 129-134.
- [5] Yanagisawa H. The material allocation problem in the steel industry[J]. *IBM J of Research and Development*, 2007, 51(3/4): 363-374.
- [6] Tang L X, Luo J X, Liu J Y. Modelling and a tabu search solution for the slab allocation problem in the steel industry[J]. *Int J of Production Research*, 2013, 51(14): 4405-4420.
- [7] Song S H. A nested column generation algorithm to the meta slab allocation problem in the steel making industry[J]. *Int J of Production Research*, 2009, 47(13): 3625-3638.
- [8] 田志波, 唐立新, 任一鸣, 等. 基于合成领域的蚁群算法求解无委托板坯匹配问题[J]. *自动化学报*, 2009, 35(2): 186-192.
(Tian Z B, Tang L X, Ren Y M, et al. Solving open-order slab matching problem by ACO with compound neighborhood[J]. *Acta Automatica Sinica*, 2009, 35(2): 186-192.)
- [9] Huang X L, Hu K Y, Li H Y, et al. Mathematical programming model and algorithms for production planning with considering slab matching[C]. *Proc of the 2nd Int Conf on Machine Learning and Cybernetics*. Xi'an: IEEE Conference Publications, 2003: 2055-2060.
- [10] Zhang T, Zhang Y J, Zheng Q P, et al. A hybrid particle swarm optimization and tabu search algorithm for order planning problems of steel factories based on the make-to-stock and make-to-order management architecture[J]. *J of Industrial and Management Optimization*, 2011, 7(1): 31-51.
- [11] Hanafi S, Freville A. An efficient tabu search approach for the 0-1 multidimensional knapsack problem[J]. *European J of Operational Research*, 1998, 106(2/3): 659-675.
- [12] Glover F. Heuristics for integer programming using surrogate constraints[J]. *Decision Sciences*, 1977, 8(1): 156-166.
- [13] Martí R, Laguna M, Glover F. Principles of scatter search[J]. *European J of Operational Research*, 2006, 169(2): 359-372.
- [14] Scheuerer S, Wendolsky R. A scatter search heuristic for the capacitated clustering problem[J]. *European J of Operational Research*, 2006, 169(2): 533-547.
- [15] Zamani R. An accelerating two-layer anchor search with application to the resource-constrained project scheduling problem[J]. *IEEE Trans on Evolutionary Computation*, 2010, 14(6): 975-984.
- [16] Russell R A, Chiang W C. Scatter search for the vehicle routing problem with time windows[J]. *European J of Operational Research*, 2006, 169(2): 606-622.
- [17] Nowicki E, Smutnicki C. Some aspects of scatter search in the flow-shop problem[J]. *European J of Operational Research*, 2006, 169(2): 654-666.

(责任编辑: 郑晓蕾)