

基于离散教与学算法的分布式预制 流水车间调度研究

曹劲松, 熊福力

(西安建筑科技大学 信息与控制工程学院, 西安 710055)

摘要: 分布式工厂生产形式对提高预制构件生产效率、保证订单按时交付、降低企业拖期交货惩罚费用具有重要的意义; 因此针对分布式预制构件流水车间调度问题, 以最小化订单总拖期惩罚为目标建立了数学优化模型, 并基于双层整数编码方式提出了一种离散教与学算法 (DTLBO); 在算法初始化阶段, 采用启发式规则和随机生成融合策略改善初始解的质量, 进而增加算法的寻优效率; 在教学阶段, 结合问题模型特点, 设计了顶层替换、底层替换两种邻域构造, 促进教师解对学生解的引导优化; 在学习阶段, 通过变异算子和交叉算子让学生解之间相互学习更新, 进一步提升算法的局部开发和全局探索能力; 试验结果表明, 与遗传算法和变邻域搜索算法对比, 提出的 DTLBO 算法具有更好的求解性能和鲁棒性; 最后与实际生产过程常用的经验启发式调度方法相比, 提出算法在目标值上表现出不低于 10% 的平均改进率, 有望显著增加预制构件制造企业净利润并提高客户满意度, 能够为企业管理者提供更佳、更合理的生产调度方案。

关键词: 教与学算法; 分布式流水车间调度; 交叉算子; 变异算子; 拖期惩罚

Research on Distributed Precast Flow Shop Scheduling Based on Discrete Teaching—Learning Based Optimization

CAO Jinsong, XIONG Fuli

(School of Information & Control Engineering, Xi'an University of Architecture and Technology,
Xi'an 710055, China)

Abstract: Distributed factory production is of great significance to improve the production efficiency of prefabricated components, ensure the on-time delivery of orders, and reduce the penalty cost of delayed delivery. Therefore, in order to minimize the penalty of total order delay, a mathematical optimization model is established for the scheduling problem of distributed prefabricated component flow shop, and a discrete teaching—learning based optimization (DTLBO) is proposed based on double-layer integer coding. In the initial stage of the algorithm, heuristic rules and random generation fusion strategy are used to improve the quality of the initial solution, so as to increase the optimization efficiency of the algorithm; in the teaching stage, combined with the characteristics of the problem model, two kinds of neighborhood structures; top-level replacement and bottom-level replacement, are designed to promote the guidance and optimization of the teacher's solution to the student's solution; in the learning stage, the mutation operator and crossover operator are used to let students learn and update each other, so as to further improve the local development and global exploration ability of the algorithm. Experimental results show that compared with genetic algorithm and variable neighborhood search algorithm, the proposed DTLBO achieves better solution quality and robustness. Finally, compared with the empirical heuristic scheduling method commonly used in the actual production process, the proposed algorithm shows an average improvement rate of no less than 10% on the target value, which is expected to significantly increase the net profit of the prefabricated component manufacturing enterprise and improve customer satisfaction, and can provide a better and reasonable production scheduling scheme for enterprise managers.

Keywords: teaching—learning—based optimization; distributed flow shop scheduling; crossover operator; mutation operator; tardiness penalty

收稿日期: 2021-04-22; 修回日期: 2021-05-25。

基金项目: 国家自然科学基金资助项目(61473216); 陕西省自然科学基金基础研究计划资助项目(2020JM-489); 陕西省重点研发计划项目(2021GY-066); 陕西省教育厅自然科学基金资助项目(17JK0459)。

作者简介: 曹劲松(1993-), 男, 安徽合肥人, 硕士研究生, 主要从事智能控制与优化调度方向的研究。

通讯作者: 熊福力(1974-), 男, 黑龙江肇东人, 硕士生导师, 博士, 副教授, 主要从事人工智能与系统优化、复杂制造系统调度优化方向的研究。

引用格式: 曹劲松, 熊福力. 基于离散教与学算法的分布式预制流水车间调度研究[J]. 计算机测量与控制, 2021, 29(12): 166-171, 183.

0 引言

流水车间调度问题 (FSSP, flow shop scheduling problem) 已被证明是一个非确定性多项式难 (NP-hard) 问题^[1-2], 问题解空间大, 复杂度高, 传统的精确算法如分支定界^[3]在求解该类问题时, 很难在合理时间内得出问题解。因此, 研究人员通常采用启发式算法对相关问题进行求解。潘子肖和雷德明^[4]针对分布式低碳并行机调度问题提出了一种基于问题性质的非劣排序遗传算法—II。Abdel 等^[5]提出了一种将鲸鱼优化算法与局部搜索策略相结合的新算法来解决置换流水车间调度问题。刘晶晶等^[6]根据柔性作业车间调度问题的特点, 以最小化完工时间为目标提出了一种混合果蝇—遗传算法, 并与遗传算法对比, 证明了所提算法的有效性。近年来, 更多的智能方法被应用于 FSSP, 如模拟退火^[7]、禁忌搜索^[8]、蚁群算法^[9]、粒子群优化^[10]、免疫算法^[11]、人工蜂群算法^[12]等。

2011 年, Rao 等^[13-15]从实际的教师教学和学生的学习过程中得到启发, 提出了一种新型的群智能算法—教与学优化算法 (TLBO, teaching-learning based optimization), 该算法具有收敛速度快, 能够屏蔽参数干扰的优点。而后, 赵乃刚^[16]将其应用在求解无约束优化问题上。马文强等^[17]设计了一种混合教与学算法有效求解了炼钢连铸调度问题。何雨洁等^[18]提出了一种混合离散 TLBO 算法求解复杂并行机调度问题。但目前关于 TLBO 算法的研究集中在求解线性化问题上, 仅少部分学者将其应用在 FSSP 领域, 且通过 TLBO 算法解决分布式预制构件流水车间调度问题的研究还未出现。

因此, 本文在充分考虑预制构件生产特殊性 (工序间的差异性较大, 工序处理周期长) 的基础上, 针对分布式预制构件流水车间调度问题, 以最小化订单总拖期惩罚为目标, 结合问题双层整数编码的特点, 提出了一种离散教与学优化算法 (DTLBO)。最后通过大量算例的测试实验, 验证了 DTLBO 算法求解问题时的有效性。

1 问题描述与建模

如图 1 所示, 分布式预制流水车间调度问题可以描述为: 来自客户的 n 个预制构件订单, 需被指派到 F 个预制构件工厂处理, 每个工厂内都有一条流水线, 流水线上的机器配置相同。生产过程由 6 道工序组成: 1) 模具组装; 2) 钢筋预埋; 3) 混凝土浇筑; 4) 蒸汽养护; 5) 模具拆除; 6) 瑕疵修整。每道工序仅有一台机器 (即单生产线工作模式)。预制构件生产不同于传统流水车间问题, 根据生产工艺的特征可将工序分为并行工序 (可同时处理多个订单) 和串行工序 (当进行到该工序下的订单未处理完前, 不可对下一订单进行处理)。其中工序 4 蒸汽养护为并行处理, 其余工序为串行处理。若订单 j 被指派到工厂 f 生产

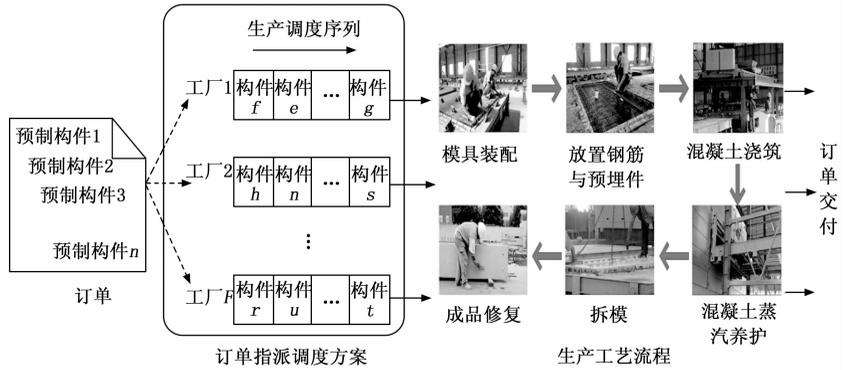


图 1 分布式预制构件生产调度系统

后, 在交货期 d_j 后完工, 则会产生拖期惩罚费用。研究以最小化总订单拖期惩罚为目标, 给出订单在工厂间最佳的指派方案和生产调度方案。

1.1 相关假设与参数说明

对于该集成优化问题的相关假设有: 1) 所有机器在调度零时刻均为可用状态; 2) 不考虑机器故障、工件损坏等突发情况; 3) 订单在工序上处理完成之前不能被其他订单抢占; 4) 订单在相邻工序间的安装和运输时间忽略不计。

表 1 调度模型参数说明

标引和输入参数:	
n	订单总数
F	工厂总数
K	工序总数
j	订单索引, $j \in n := \{1, 2, \dots, n\}$
f	工厂索引, $f \in F := \{1, 2, \dots, F\}$
k	工序索引, $k \in K := \{1, 2, \dots, 6\}$
$p_{j,k}$	订单 j 工序上的处理时间
d_j	订单 j 的交货期
β_j	订单 j 的单位时间拖期惩罚费用
决策变量:	
$x_{j,f}$	如果订单 j 被指派到工厂 f 为 1, 否则为 0
$y_{i,j,f}$	在工厂 f 内, 如果订单 j 在订单 i 后立即加工为 1, 否则为 0
$C_{j,f}$	订单 j 在工序 k 上的完成时间

1.2 数学模型

基于以上符号和假设建立数学模型如下:

$$\min \sum_{f=1}^F \sum_{j=1}^n \beta_j \cdot x_{j,f} \cdot \max\{C_{j,6} - d_j, 0\} \quad (1)$$

s. t.

$$\sum_{f=1}^F x_{j,f} = 1, \forall j \quad (2)$$

$$\sum_{j \in n \setminus \{i\}} y_{i,j,f} \leq x_{j,f}, \forall f, i \in n \cup \{0\} \quad (3)$$

$$\sum_{i \in n \setminus \{j\}} y_{i,j,f} = x_{j,f}, \forall f, j \quad (4)$$

$$C_{j,k} \geq C_{j-1,k} + P_{j,k}, \forall j, k \in K \setminus \{4\} \quad (5)$$

$$C_{j,k} \geq C_{j,k-1} + P_{j,k}, \forall j, k \in K \setminus \{4\} \quad (6)$$

$$C_{j,k} = C_{j,k-1} + P_{j,k}, \forall j, k \in \{4\} \quad (7)$$

$$C_{j,k} \geq 0 \quad (8)$$

式 (1) 为目标函数最小化订单总拖期惩罚费用; 式 (2) 确保每个订单被指派到某一工厂中; 式 (3) 和式 (4) 表示订单 j 是否和订单 i 指派到同一工厂, 且订单 j 是否在订单 i 紧后加工; 式 (5) 表示订单在工序上的处理时间不早于上一订单在此工序上的完工时间, 第 4 道工序蒸汽养护为并行处理过程, 不需满足此约束; 式 (6) 表示订单在工序上的开始处理时间不早于该订单上一工序的完工时间; 式 (7) 为并行工序完工时间的计算公式; 式 (8) 定义了决策变量的取值范围。

2 离散教与学优化算法

TLBO 算法原理^[14] 是模拟以班级为单位的学习方式, 通过教师的“教”来引导班级学生成绩提高; 同时, 学生之间通过相互“学习”的方式进一步促进知识的吸收。其中, 教师和学生相当于进化算法中的个体, 而教师是对应目标值最佳的个体, 学生成绩提高表示优化目标得到改进。TLBO 算法算法分为 3 个阶段: 1) 教师阶段: 在班级中寻得最优解个体作为教师, 引导班级中的学生搜索更好的解; 2) 学生阶段: 班级中学生通过与其他学生学习, 获取比自身更好的解; 3) 角色互换: 在某次迭代中, 若有学生寻得的解比教师的解更好, 则在下一次迭代中将该学生替换为教师, 教师降级为学生。

针对分布式预制构件流水车间调度该类非线性整数规划问题, 本文在 TLBO 算法的基础上提出了一种离散教与学优化算法 (DTLBO), 算法步骤如下:

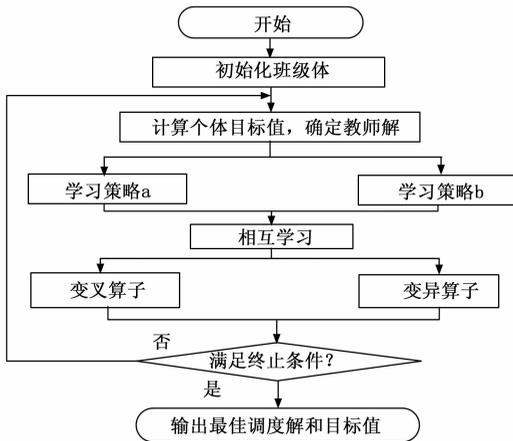


图 2 DTLBO 算法框图

Step1: 通过结合启发式策略和随机生成的方式生成初始“班级”群体 pop , 确定教师解 X_t ;

Step2: 教学阶段, 通过随机地选择学习策略 a 和学习策略 b 两种交叉算子, 将每个学生个体解 X_i 与教师解 X_t 进行信息交互, 若更新后的学生个体解 X_i^{new} 优于 X_i , 则将 X_i^{new} 替换为 X_i ;

Step3: 学习阶段, 学习阶段分为相互学习和自我学习,

相互学习过程是随机将学生个体两两配对, 进行与步骤 2 相同的信息交互操作, 若得到的解质量更好则更新学生个体;

Step4: 自我学习过程是随机地选择变异算子和交叉算子对每个学生个体 X_i 进行更新操作得到 X_i^{new} , 若 X_i^{new} 优于更新前 X_i , 则将 X_i^{new} 替换为 X_i ;

Step5: 将学生个体 X_i 中目标值最优的个体 X^* 取出与教师解 X_t 比较, 如对应目标值更佳, 则将 X^* 替换为 X_t ;

Step6: 判断是否满足终止条件: 否, 返回步骤 2; 是, 输出最终教师解 X_t 和对应目标值。

2.1 调度解编码解码方式

以 10 订单和 3 预制构件工厂的问题为例, 如图 3 所示, 从区间 $[1, F]$ 随机生成 n 个数字作为解的第一层, 从区间 $[1, n]$ 生成 n 个不重复的数字作为解的第二层。一层编号为 (1, 2, 1, 3, 2, 3, 1, 2, 3, 2) 表示订单到工厂的指派方案, 二层编号为 (8, 7, 3, 2, 5, 6, 10, 1, 9, 4)。即订单 (8, 3, 10) 被分配至工厂 1 处理; 订单 (7, 5, 1, 4) 被分配至工厂 2 处理; 订单 (2, 6, 9) 被分配至工厂 3 处理。订单在厂内的处理顺序如括号内从左到右所示, 如此确定了订单在不同工厂间的指派方案以及生产调度方案。

1	2	1	3	2	3	1	2	3	2
8	7	3	2	5	6	10	1	9	4

图 3 编码示意图

2.2 初始化“班级”群体

为了避免随机方式产生的初始解群体质量较差, 设计了一种启发式策略, 即按各订单交货期排序, 再将每个待加工订单依次分配至各工厂, 基于该策略生成一个班级个体, 其余班级个体通过随机方式生成。通过启发式规则和随机生成融合策略提升初始解的质量, 来增加算法的寻优效率。

2.3 教学阶段

在教学阶段, 结合问题模型双层编码结构的特点, 设计了两种学习策略, 通过让班级中每个学生个体随机选择两种学习策略中的一个与教师进行信息交互, 提高班级学生的成绩。两种学习策略内容如下:

1) 学习策略 a: 如图 4 所示, 将该次迭代中的教师解 X_t 和学生个体 X_i 取出, 从区间 $[1, n]$ 随机选取两个位置点, 将教师解一层被选位置的值替换到学生解第一层的被选位置, 从而得到经过教学后的学生个体 X_i^{new} 。

2) 学习策略 b: 如图 5 所示, 从区间 $[1, n]$ 随机选取两个位置点, 确定教师解 X_t 第二层对应被选位置的值, 找出学生解 X_i 第二层数值与教师解被选值相同的位置作为被选位置, 最后将教师解中的被选值按序替换到学生解第二层的被选位置, 得到经过教学后的学生个体 X_i^{new} 。

2.4 学生阶段

2.4.1 相互学习

将学生个体两两随机配对, 进行与教学阶段相同的信

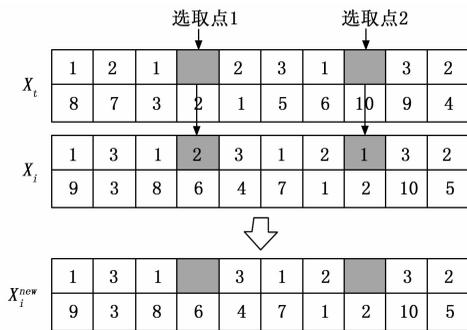


图 4 顶层替换

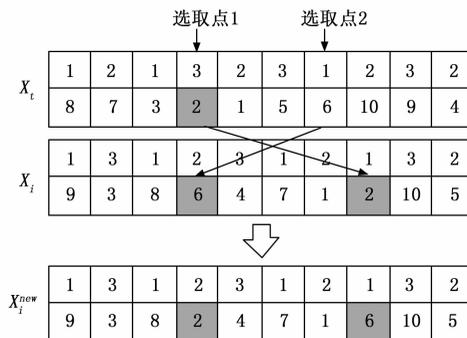


图 5 底层替换

息交互操作, 若得到的解质量比相互学习前更好则更新学生个体。

2.4.2 自我学习

本文结合问题模型的特点, 设计了两种自我学习过程:

1) 如图 6 所示, 将每个学生个体 X_i 取出, 通过变异算子在解的第一层随机选取两个位置点, 然后从 $[1, F]$ 内随机产生和选取点不一样的值进行替换, 从而改变订单在工厂间的指派。

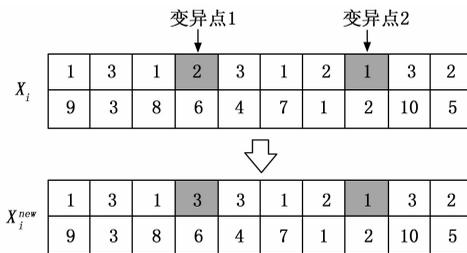


图 6 变异算子

2) 如图 7 所示, 将每个学生个体 X_i 取出, 通过交叉算子在解中随机选取两个位置点, 将两位置点一二层包含的元素整层互换, 从而改变订单的生产调度排序。

2.5 参数设置

为了评估 DTLBO 算法的性能, 将其与生产调度领域常用的智能算法 GA^[19] 和 VNS^[20] 作对比。公平起见, 3 种算法的终止准则统一设置为最大运行时间 $t = n * 6 / 10$ 。需要调节的参数为 DTLBO 的种群大小 (pop), 以及 GA 算法的种群大小 (pop)、交叉率 (p_c) 和变异率 (p_m)。参数调整结

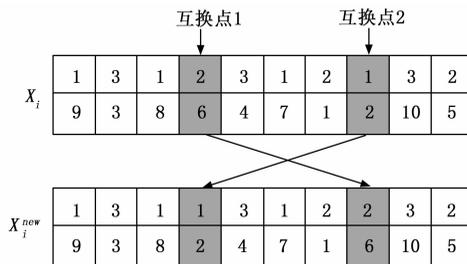


图 7 交叉算子

果如下:

- 1) DTLBO: $pop = 100$;
- 2) GA: $pop = 80, p_c = 0.9, p_m = 0.1$ 。

3 分布式预制生产调度系统

3.1 系统原理

分布式预制生产调度系统的原理是: 先将设定好的算法导入上位机, 上位机接收到来自客户的订单信息, 包括订单数量、构件类型、交货期以及各类型构件延期交付的惩罚费用; 然后上位机根据订单信息计算出最佳的订单指派和调度方案; 最后通过下位机将调度方案显示反馈给生产管理人员, 管理人员依据给出的调度序列控制各预制构件订单的开工时间完成调度工作。

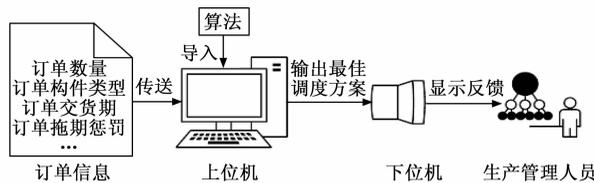


图 8 系统原理图

3.2 调度过程

以 2.1 节给出的 10 订单调度解为例, 首先对其进行解码, 得出最终可指导实际生产的调度方案, 如表 2 所示。

表 2 调度方案

调度解	调度方案
1 2 1 3 2 3 1 2 3 2	工厂 1: 8→3→10
	工厂 2: 7→5→1→4
8 7 3 2 5 6 10 1 9 4	工厂 3: 2→6→9

根据该调度方案将各订单依次指派到对应的工厂按序加工处理。以工厂 1 为例, 首先在工序 1 上处理订单 8, 由式 (5) ~ 式 (7) 计算订单 8 在工序 1 上的完成时间, 并作为其紧后处理的订单 3 在工序 1 上的开工时间, 再通过式 (5) ~ (7) 计算订单 3 在工序 1 上的完成时间, 依次类推计算得出预制工厂 1 内生产调度序列 (8 3 10) 内所有订单在 1~6 工序上的最佳开工时间、完工时间。生产管理者通过控制各订单的开工时间完成对订单的生产调度, 并且已知各订单最终的完工时间和各订单的交货期, 带入目标函数式 (1) 即可算出总订单拖期惩罚费用。图 9 为厂 1 内预

制构件订单生产调度甘特图，该图反应了经算法优化后的各订单在每道工序上的最佳开工时间及完工时间。

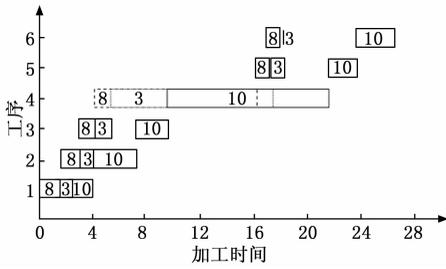


图 9 生产调度甘特图

表 3 订单生产信息

类型	每道工序加工时间/小时						拖期 惩罚 系数
	模具 装配	钢筋 绑扎	浇筑	养护	脱模	修整	
1	2.0	1.6	2.4	12	2.5	1.0	10
2	3.4	4.0	4.0	12	2.4	5.0	10
3	0.8	1.0	1.2	12	0.8	0.0	10
4	0.6	0.8	1.0	12	0.6	2.0	10
5	3.0	3.6	2.4	12	2.4	3.0	10
6	3.0	3.2	3.0	12	3.0	1.6	10
7	1.3	0.9	2.4	12	1.9	1.8	10
8	1.7	1.4	1.1	12	0.9	0.7	20
9	2.2	1.8	1.2	12	2.3	0.7	20
10	1.6	3.2	2.3	12	2.1	2.7	20

4 仿真实验与结果分析

实验通过 Matlab 2018b 编程实现，计算机配置为 Microsoft Windows 10，处理器为 Intel Core i5-9400F CPU @ 2.9 GHz/8 GB RAM。通过不同规模下的算例实验，对各算法求解分布式预制构件调度问题时的鲁棒性和求解性能进行比较分析。

4.1 测试实例

如表 3 所示，每种类型订单各工序的处理时间取自 Brandimarte 等^[21]文献中的标准数据，订单的单位时间拖期惩罚系数 $\beta_i \in \{10, 20\}$ ，问题实例的交货日期 d_i 在区间 $[P_T \cdot n, 3 * n]$ 均匀分布生成， P_T 为所有订单处理时间的累加和。实验中设定预制构件工厂数量为 3，考虑了 20、30 和 50 三种待调度订单数量，在每种数量规模下测试了 5 个问题实例，每个问题实例运行 20 次，记录 20 次运行取得目标值的最小值 (Min)、平均值 (Avg) 和标准差 (Std)。

4.2 算法性能对比分析

为了直观地比较 3 种算法性能的优劣。本文采用相对偏差率 (RPD, relative percentage deviation) 来评估各算法的性能。计算公式如下：

$$RPD_{n,m}^A = \frac{f_{n,m}^A - f_{n,m}^{best}}{f_{n,m}^{best}} \times 100\% \quad (11)$$

$$ARPD_n^A = (\sum_{m=1}^5 RPD_{n,m}^A) / 5 \quad (12)$$

其中： $f_{n,m}^{best}$ 是订单数量为 n 时，问题实例 m 通过各算法求得目标函数的最佳值； f_m^A 为问题实例 m 通过算法 A 求得的值； $RPD_{n,m}^A$ 是订单数量为 n 时，算法 A 求解问题实例 m 计算出的偏差率； $ARPD_n^A$ 为算法 A 在计算订单数量为 n 的问题实例时，计算出的平均偏差率。由此可见，RPD 值、ARPD 值越小说明算法求解性能越好。

4.3 实验结果分析

表 4 记录了各算法在计算 20、30 和 50 三种不同待调度订单数量问题实例下得出的最小 RPD 值、平均 RPD 值和 Std 值。为了便于区分，对同一个问题实例各算法得出的最小 RPD 值用粗体表示，最小平均 RPD 值用粗斜体表示，最小 Std 值用斜体表示。

从表 4 可以看出，实验测试了不同订单规模下的 15 个

表 4 实验测试结果对比

实例	最优拖期	GA			VNS			DTLBO		
	目标值	<i>Min</i>	<i>Avg</i>	<i>Std</i>	<i>Min</i>	<i>Avg</i>	<i>Std</i>	<i>Min</i>	<i>Avg</i>	<i>Std</i>
20_1	4 416	10.52	14.28	203.7	5.80	9.81	106.6	0.00	1.38	47.8
20_2	4 381	7.56	12.29	161.7	2.35	5.24	94.1	0.00	0.95	30.7
20_3	4 520	8.53	10.19	283.7	5.82	7.51	61.4	0.00	1.07	44.0
20_4	4 538	9.03	12.98	281.8	6.46	10.04	104.1	0.00	1.27	36.3
20_5	4 230	11.02	14.77	92.1	5.53	9.44	139.2	0.00	2.83	69.2
30_1	9 140	8.60	13.82	493.4	4.40	5.65	93.3	0.00	2.04	142.2
30_2	9 578	12.32	15.27	345.4	7.62	10.02	181.4	0.00	0.81	70.2
30_3	8 894	14.02	15.99	498.0	4.68	6.35	168.6	0.00	1.52	96.8
30_4	9 798	13.28	16.32	306.1	3.02	4.69	139.4	0.00	1.69	109.7
30_5	8 758	13.27	16.98	448.6	3.25	5.02	113.1	0.00	0.96	91.4
50_1	26 195	13.79	15.67	701.9	7.23	7.62	137.6	0.00	0.75	169.3
50_2	27 620	11.37	14.44	846.9	7.47	9.00	244.1	0.00	0.47	138.4
50_3	27 244	14.00	16.87	607.6	6.16	8.07	407.6	0.00	1.07	163.4
50_4	28 537	10.19	11.97	625.4	3.40	4.65	260.8	0.00	0.56	124.8
50_5	25 791	12.56	15.32	839.8	2.69	4.19	578.4	0.00	0.66	161.7

算例, 对于各算例下的最佳平均 RPD 值和最小 RPD 值, 均由 DTLBO 算法取得。各算例最小的 Std 值, DTLBO 算法取得了 14 个。图 10 是 3 种算法在求解不同订单规模问题时的 $ARPD$ 值对比图, 可以看出在 3 种订单数量问题下, 本文提出的 DTLBO 算法均取得了最佳 $ARPD$ 值。即表明, 与 GA 和 VNS 相比, 提出的 DTLBO 算法具有更好的求解性能和鲁棒性。

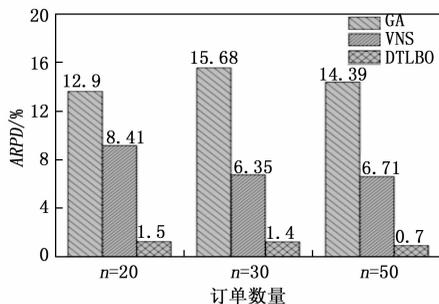


图 10 各订单数量的平均偏差率图

以 30 订单规模下的算例 5 为例, 图 11 为 3 种算法的迭代收敛曲线, 可以看出 DTLBO 算法不仅有较好的求解质量, 且收敛速度最快。

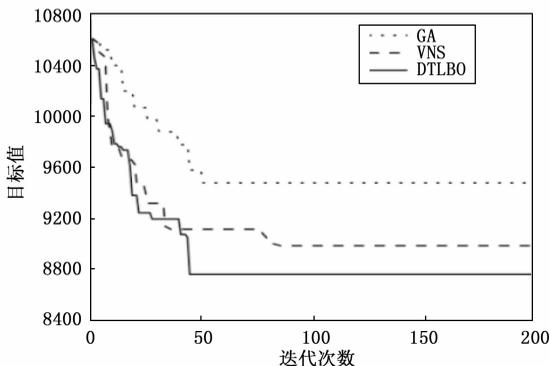


图 11 收敛曲线

4.4 与启发式调度方法对比

传统预制构件企业通常采用基于经验的启发式调度方法: 即先将订单按交货期的早晚排序, 基于贪婪的思想将交货期早的订单置于调度序列的前部, 先进行加工处理, 如此依次将各个订单分配至所有工厂内, 得出调度方案。

为了验证本文提出算法的实际应用价值, 表 5 统计了 3 种订单规模下分别通过 DTLBO 算法和启发式调度方法计算得出的平均目标值, DTLBO 算法较启发式调度方法在目标值上的改进率分别为 11.2%、10.8% 和 12.4%。可以看出, 使用 DTLBO 算法提供的调度方案可以为企业有效降低订单拖期惩罚费用成本, 增加预制构件制造企业净收益。

表 5 不同订单规模下 DTLBO 算法的改进率

n	20	30	50
启发式调度方法	4 984.7	10 374.6	30 648.0
DTLBO	4 482.7	9 363.4	27 267.0
改进率/%	11.2	10.8	12.4

5 结束语

本文针对分布式预制构件流水车间调度问题, 以最小化订单总拖期惩罚为目标构建了非线性整数规划数学模型, 并根据模型特点设计了双层整数编码形式, 最后提出了一种离散教与学优化算法 (DTLBO) 应用于求解该问题。通过大量仿真实验, 从最优 RPD 值、平均 RPD 值和 Std 值 3 个性能指标与 VNS 算法和 GA 算法对比分析, 结果验证了提出的 DTLBO 算法在求解分布式预制构件流水车间调度问题时的有效性和可行性。在下一步研究中, 可以将机器故障、临时订单等突发情形集成到调度问题中进行求解。

参考文献:

- [1] RUIZ R, STUTZLE T. A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem [J]. European Journal of Operational Research, 2007, 177 (3): 2033-2049.
- [2] RIBAS I, COMPANYS R, TORT-MARTORELL X. An iterated greedy algorithm for the flowshop scheduling problem with blocking [J]. Omega, 2011, 39 (3): 293-301.
- [3] WANG X, XIE X, CHENG T C E. Order acceptance and scheduling in a two-machine flowshop [J]. International Journal of Production Economics, 2013, 141 (1): 366-376.
- [4] 潘子肖, 雷德明. 基于问题性质的分布式低碳并行机调度算法研究 [J]. 自动化学报, 2020, 46 (11): 2427-2438.
- [5] ABDEL-BASSET M, MANOGARAN G, EL-SHAHAT D, et al. A hybrid whale optimization algorithm based on local search strategy for the permutation flow shop scheduling problem [J]. Future Generation Computer Systems, 2018, 85: 129-145.
- [6] 刘晶晶, 刘业峰, 黎虹. 基于混合果蝇-遗传算法求解柔性作业车间调度问题 [J]. 计算机测量与控制, 2020, 28 (12): 227-232.
- [7] LOW C. Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines [J]. Computers & Operations Research, 2005, 32 (8): 2013-2025.
- [8] WANG S, LIU M. Two-stage hybrid flow shop scheduling with preventive maintenance using multi-objective tabu search method [J]. International Journal of Production Research, 2014, 52 (5): 1495-1508.
- [9] ALAYKYRAN K, ENGIN O, DOYEN A. Using ant colony optimization to solve hybrid flow shop scheduling problems [J]. The international journal of Advanced Manufacturing Technology, 2007, 35 (5): 541-550.
- [10] WANG D, TAN D, LIU L. Particle swarm optimization algorithm: an overview [J]. Soft Computing, 2018, 22 (2): 387-408.
- [11] JIANG Y, HAO K, CAI X, et al. An improved reinforcement-immune algorithm for agricultural resource allocation optimization [J]. Journal of Computational Science, 2018, 27: 320-328.

(下转第 183 页)