

Scheduling with Multi-Attribute Preparation Times on Two Parallel Machines

Hui Xiong^{1,2*}, Peifeng Liang¹, Xiaojia Wang¹, Bin Zhu¹

¹Guangdong Guangye Kaiyuan Technology Co., Ltd., Guangzhou Guangdong

²Guangzhou Close-Watching Education Information Co., Ltd., Guangzhou Guangdong

Email: *375610596@qq.com

Received: Oct. 7th, 2018; accepted: Oct. 22nd, 2018; published: Oct. 30th, 2018

Abstract

The scheduling problem with multi-attribute preparation times studied here is from the polyvinyl chloride leather (PVCL) plant. On two identical parallel machines, the products are processed in sequence of different levels, attributes, and jobs. The objective is to determine a schedule for two identical parallel machines to minimize the makespan. A constructive heuristic is firstly proposed and evaluated by comparing with the current scheduling method used by the case plant. To further improve the solution, a variable neighborhood search (VNS) meta-heuristic is presented and compared with a mixed integer programming model. The computational results show that the heuristic outperforms the current scheduling method with a significant improvement, and the VNS can further improve the solution.

Keywords

Parallel Machines, Scheduling Problem, Multi-Attribute Preparation Times, Heuristic Algorithm, Variable Neighborhood Search

两台并行机器上进行多属性准备时间的工序设置问题

熊辉^{1,2*}, 梁培锋¹, 王晓佳¹, 朱彬¹

¹广东广业开元科技有限公司, 广东 广州

²广州临观教育信息有限公司, 广东 广州

Email: *375610596@qq.com

收稿日期: 2018年10月7日; 录用日期: 2018年10月22日; 发布日期: 2018年10月30日

*通讯作者。

摘要

本文研究聚氯乙烯皮革(PVCL)工厂的多重属性准备时间的工序设置问题。针对于两台同样的并行机器,按顺序生产具有不同层次、不同属性、不同工种的PVCL制品,目标函数是设置一个加工排序,使得最大完工时间极小化。首先提出一个建构式的启发式演算法,与工厂现行排序法做比较,以评估此算法的绩效;继续提出变邻域搜寻算法,以进一步改善启发式算法所求得解,并与混合整数规划模型做比较。实验结果证明,本文的启发式演算法显著性地优于现行PVCL工厂所使用的排序方法,而VNS算法的运行结果也证实能有效改善启发式演算法所求得解。

关键词

并行机器, 工序设置, 多重属性准备时间, 启发式演算, 变邻域搜索

Copyright © 2018 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 绪论：问题描述与分析

本文研究一个与聚氯乙烯皮革(Polyvinyl Chloride leather, PVCL)的实际生产相关的时序安排问题,其来自我们所调研的某一家工厂,而且已成为行业内普遍急需解决的问题。PVCL 的应用领域很广泛,主要涉及: 1) 鞋类: 足球鞋, 运动器材, 工作鞋, 凉鞋和雪地靴; 2) 家具: 沙发, 办公椅, 按摩椅, 休闲和温泉浴场的遮挡物; 3) 海运: 快艇和小船的内部结构(船座和内部); 4) 运输工具: 车座, 门板和扶手椅; 5) 运动设备: 自行车, 手套和辘重的配件; 6) 医药护理: 医护用的椅子和床。

PVCL 是一个资本高度集中的产业,为了开发使用这种设备,工厂每周七天工作,每八小时轮换一次,每天轮换三次。PVCL 工厂的主要设备是皮革轧光机,但这种机器却是这个系统中的瓶颈。皮革轧光机是一组强压轮,用来使产品表面平滑,同时让它们的厚度、斑纹和宽度都相同。PVCL 系统的运作能力主要取决于皮革轧光机,而且它的生产力不能靠投入更多的人力和时间来提高。工厂每月的平均生产量大概为 600,000 码,换句话说,它每分钟的产出率为 14 码。每一个采购订单都包含 PVCL 的许多产品,每个产品都应具备五种属性: 斑纹、硬度、颜色、厚度以及宽度;每一个性质又包含了几种不同的层次和相应的属性准备时间。采购订单上的一个 PVCL 产品可以被看成是一项时序操作工种。由于没有时间设置的约束,五个产品属性都处于同一层次水平的 PVCL 产品被归为同一类工种。因为在两组相近的工种中至少有一个特质的层次不同,因此为了处理工种之间的转换问题,有必要在皮革轧光机上安装一个装备调节器。例如,两组相近的产品在厚度和颜色上有差异,就有必要在皮革轧光机上对厚度和颜色进行调节,其准备时间等于厚度和颜色属性准备时间之和。因此,两组相近产品的准备时间是由那些不同层次的属性准备时间的总和来决定的。每一类 PVCL 皮制品的加工时间通常为 3~15 个小时,每周进行一次产量计划。

我们所研究的工厂配备八个皮革轧光机,各台轧光机通常负责生产不同的皮制品。但有时一两台机器会因没有相关的采购订单而闲置。为了避免这种情况,同时提高轧光机的使用率,当一台机器被闲置时,工厂会计划安排这台机器去协助另外一台采购订单最多的机器。所以,在工厂里经常会有单一机器

作业和两台并行机器同时作业的情况。在本研究中，我们关注的就是两台相同的并行机器共同作业的情况，我们将其界定为两台相同并行机器的工序设置问题。

对于 PVCL 生产系统，Abreu 等[1]、Chen [2]与 Lee 等[3]曾在单一机器上用从属序列的多属性准备时间来处理时序安排问题，其中，Lee *et al.* [1]在既有的理论上提出一个构造性的启发式算法。计算结果表明，相比于 PVCL 薄板工厂之前所使用的时序安排方法，该构造性的启发式算法具有显著性的改进。在文[4] [5]中，虽然也研究了并行机器的排序问题，但是机器之间是不相关的；文[6] [7] [8] [9]虽然讨论的是相同的并行机器处理问题，但所研究案例工厂与本文的问题差异较大。在本文中，我们将提出一个简单而有效的启发式安排方法以满足 PVCL 工厂的要求，并将其与工厂当前所使用的时序安排方法进行比较，从而评估我们所提出的启发式方法的效力和效率。为了进一步完善启发式的结果，我们还提出了一个变邻域搜索启发式算法，其也将与我们所提出的启发式方法及混合整数规划模型进行比较，以显示其性能。

假设两台相同的并行机器分别用 M_1 和 M_2 表示，第 i 类工种 J_i 的处理时间为 P_i ，且 J_i 可能是被两台机器中的任一处理， $i=1,2,\dots,n$ 。此外，任何工作都不具有优先处理权。任意两个相邻工种之间必须间隔一个准备时间。所有的机器都可在零点开始处理工种，而此时所有的工种都应准备就绪。生产过程不允许中断且任何工种都不具有优先权。机器每次至多只能进行一种工种，一种工种不能同时在两台机器上进行。这样假设的目的是为了尽量减小最大的完成时间，即完成时间。之所以将完成时间按最小准则选取，是因为它与目标有重要的利害关系，且其确保两台并行机器保持负载均衡，Pinedo [10]对这一思想有详尽的描述。设 C_i 为 J_i 的执行时间，则目标为

$$\min C = \max_{1 \leq i \leq n} C_i$$

根据这三组符号，问题可以由 P2/ s_{ik} /C 表示。其中 P2 指定两台相同的并行机器， s_{ik} 代表从属序列准备时间，C 表示完成时间， $i,k=1,2,\dots,n$ 。当一台机器由生产工种 J_i 转换为工种 J_k 时，需要从属序列准备时间 $s_{ik} > 0$ ；当 $s_{ik} = s_{ki}$ 时，准备时间模型是对称的。不失一般性，设 $s_{ii} = 0$ ，即同类工种之间可自然衔接，不需要额外的准备时间。

由 Pinedo [10]可知，单一机器的从属序列准备时间安排(P1/ s_{ik} /C)的极小化问题是 NP 难题，故 P2/ s_{ik} /C 也是个 NP 难题。对于并行机器的从属序列准备时间问题已有现存的启发式，如文献[11] [12] [13]，但它们都没有直接地解决并行机器多属性次序安排这种问题。由于现实生活中的调度者很容易将注意力集中于较长的属性准备时间，那么在安排时保持这种多属性次序是有利的。调度者并不想失去多属性的特征，他们需要解决这个安排问题的简单方法。因此，从简化和实用的角度出发，我们需要研究一个简单的方法来解决多属性并行机器的时间设置问题。

2. 工厂现有的安排方法

如表 1 所示，为 PVCL 厂提供的标准属性准备时间：

Table 1. The preparation time of standard property in PVCL factory

表 1. PVCL 厂的标准属性准备时间

属性	斑纹	颜色	硬度	宽度	厚度
准备时间(分钟)	60	15	20	15	10

在作为案例的工厂中，调度者为了处理本文所提出的问题，需要考虑两个方面。其一，为平衡负载，应决定哪些工种被指定到哪台轧光机上处理；其二，确定每台轧光机的工种序列，以减少准备时间。在

这基础上, 计划员首先采用最长处理时间优先(longest processing time first, LPTF)规则, 分别给两台轧光机分配耗时最长的两个工种; 然后, 当一台轧光机可用时, 再按此原则继续分配。LPTF 规则试图将耗时最短的工种安排在最后, 以便能平衡工作量。在 P2 的情况中, 不考虑准备时间而使用 LPTF 规则就能得到很好的效果。接着, 调度者为了减少每台轧光机所用的准备时间, 试图同时安排它们生产更多同一层次的工种。斑纹是最重要的属性, 有着最长的属性准备时间(见表 1), 因此, 调度者想要尽可能地缩短其准备时间。他们认为最重要的是在轧光机上先确定五个属性的优先权。综合以上的信息, 可知安排步骤大致为: (1) 将两个生产时间最长的工种分别分配给轧光机 M_1 和 M_2 , 当其中一台完成时, 再将剩余工种中耗时最长者分配给它, 重复这个操作, 一直到完成所有工种。(2) 在属性时段非递增的情况下, 确定属性的优先顺序为: 斑纹, 硬度, 宽度, 颜色和厚度。(3) 在每台轧光机上, 将斑纹相同层次的工种归成一组, 在斑纹层次非递减的情况下, 将一组组工种按顺序排好。(4) 对于斑纹层次相同的工种, 按硬度层次非递增地进行排列。(5) 根据步骤(2)的属性最优顺序, 参照步骤(4), 执行排序步骤, 一个个地进行排列。工种序列满足颜色层次非递减, 宽度和厚度层次非递增。

如表 2 所示, 为案例工厂所提供的 11 个工种, 我们用这个实例解释当前的安排方法。

Table 2. 11 examples of jobs

表 2. 11 个工种的实例

属性	工种											属性时段
	1	2	3	4	5	6	7	8	9	10	11	
斑纹(编号)	001	191	001	001	002	002	001	191	002	001	001	60
宽度(英寸)	48	62	42	48	48	52	42	30	50	50	50	15
厚度(毫米)	0.8	0.8	1.0	1.0	0.8	0.8	0.8	0.8	1.2	1.2	0.8	10
硬度(编号)	8	8	16	9	7	7	8	8	10	10	9	20
颜色(编号)	9	1	2	1	1	1	2	2	9	13	2	15
P_i (分钟)	550	178	488	860	675	326	745	204	618	842	723	分钟

在步骤(1)中, 应用 LPTF 规则, 根据 P_i , 可以得到序列

$$M_1 = (J_4, J_{11}, J_5, J_3, J_2), M_2 = (J_{10}, J_7, J_9, J_5, J_8) \quad (2.1)$$

其中准备时间还未被考虑在内; 在步骤(2)中, 属性的优先顺序由以下方面决定: 斑纹, 硬度, 颜色, 宽度和厚度; 在步骤(3)里, 根据斑纹的相同层次, 机器 M_1 把工种 J_4, J_{11}, J_3 归为一组, 斑纹编号为 270; 工种 J_5, J_6 为一组, 斑纹编号为 507; 工种 J_2 自成一组, 斑纹编号为 663。而机器 M_2 将 J_{10}, J_7, J_1 归为一组, 斑纹编号为 270; J_9 自成一组, 斑纹编号为 507; J_8 自成一组, 斑纹编号为 663。在步骤(4)中, 对于在机器 M_1 , 根据硬度层次 16 和 9 的非递增顺序, 可以得到序列 J_3, J_4, J_{11} ; 同样地, 其他工种组可以得到序列 J_5, J_6 与自成一组的 J_2 。用相同的方法, 机器 M_2 可以得到与机器 M_1 类似的序列 J_{10}, J_7, J_1 , J_9 自成一组, J_8 也自成一组。对于宽度, 颜色和厚度, 继续执行这个安排操作, 得到最后的计划表序列为

$$M_1 = (J_3, J_{11}, J_4, J_6, J_5, J_2), M_2 = (J_{10}, J_1, J_7, J_9, J_8) \quad (2.2)$$

完成时间为 $C = 3367$ 分钟。

3. 混合整数规划模型

解决 P2/ s_{ik}/C 的最佳方法之一是列出其数学模型, 然后用优化软件对其进行求解。根据问题中变量的性质, 我们提出一个混合整数规划(mixed integer programming, MIP)模型。设 L 为一个非常大的数, 且二

值决策变量。

$$y_{im} = \begin{cases} 1, & \text{工种 } i \text{ 在机器 } m \text{ 上处理,} \\ 0, & \text{其他情况;} \end{cases} \quad (3.1)$$

$$x_{ij} = \begin{cases} 1, & \text{同一机器上工种 } i \text{ 后接着处理工种 } j, \\ 0, & \text{其他情况;} \end{cases} \quad (3.2)$$

则 P2/ s_{ik}/C 问题的 MIP 模型为

$$\min Z = C; \quad (3.3)$$

$$\text{s.t. } \sum_{m=1}^2 y_{im} = 1, \quad i = 1, \dots, n \quad (3.4)$$

$$y_{im} + (1 - y_{jm}) + x_{ij} \leq 2, \quad i = 1, \dots, n-1; j = i+1, \dots, n \quad (3.5)$$

$$C_j - C_i + L(3 - x_{ij} - y_{im} - y_{jm}) \geq P_j + s_{ij}, \quad i = 1, \dots, n-1; j = i+1, \dots, n \quad (3.6)$$

$$C_i - C_j + L(2 - x_{ij} - y_{im} - y_{jm}) \geq P_j + s_{ji}, \quad i = 1, \dots, n-1; j = i+1, \dots, n \quad (3.7)$$

$$C_i \geq P_i y_{im}, \quad i = 1, \dots, n; m = 1, 2 \quad (3.8)$$

$$Z \geq C_i, \quad i = 1, \dots, n \quad (3.9)$$

目标(3.3)是最小化完成时间, 约束条件(3.4)确保每个工种被安排到某一台机器上; (3.5)确保仅当工种 i 和 j 被同时安排到同一机器上时, 其优先问题才需要被考虑; 若工种 i 和 j 被安排到不同的机器上生产, 则根据(3.2)可知, 那么 $x_{ij} = 0$; 当工种 i 和 j 被同时安排到同一台机器上, 条件(3.6)和(3.7)为它们的处理完成时间建立了联系; 仅当 $y_{im} = 1$ (即工种 i 在机器 m 上生产)时, 条件(3.8)才有意义, 并且决定它们的生产时间; 若 $y_{im} = 0$, 则条件(3.8)是多余的; 条件(3.9)决定目标值。

虽然 MIP 模型是解决小型问题的最好方法之一, 但对于中型和大型问题, 普通的优化软件却不能从实际上将其解决, 即难以得到全局最优解。因此, 我们需要高效的算法来解决这个问题。在下一部分, 我们将提出一个启发式算法, 并进一步提出变邻域搜索(variable neighborhood search, VNS)元启发式方法, 用以完善我们所提出的启发式算法得到的结果。

4. 启发式算法

4.1. 相邻处理时间指数

要解决并行机器准备时间的时序安排问题, 需要考虑两个因素: 平衡机器的工作量和减少每台机器的准备时间。为了达到这两个目的, 我们引入了启发式发展指数。我们将最小相邻指数第一定理[3]和处理时段结合成一个指数, 称之为相邻处理时间指数(Adjacent Processing Time Index, APTI), 而工种 J_i 的 APTI 记为 $APTI(J_i)$, 数学上定义为

$$APTI(J_i) = \frac{1}{P_i} \sum_{k=1}^{|A|} [A_k \cdot N_k(J_i)] \quad (4.1)$$

P_i 是工种 J_i 的处理时间, $|A|$ 为属性种数(此处 $|A| = 5$), A_k 表示第 n 个属性准备时间, $N_k(J_i)$ 表示与工种 J_i 的第 k 种属性同层次工种的数量。例如, 在表 2 中, 对于 J_1 , 有五个斑纹层次一致的工种(即 270), 两个宽度层次一致(即 48), 六个厚度层次一致的工种(即 0.8), 三个硬度层次一致的工种(即 8), 和一个颜色层次一致的工种(即 9)工种。因此, 有

$$APTI(J_1) = \frac{1}{550} (60 \times 5 + 15 \times 2 + 10 \times 3 + 15 \times 1) = 0.8455$$

APTI 同时涉及到处理时间, 工作灵活度和准备时间。在并行机器的环境下, 平衡工作量对分配工作来说非常重要。平衡并行机器的工作量时, 前文所介绍的最长处理时间优先(LPTF)原则无法达到理想的结果。但为了突出 LPTF 的这个特点, 我们没有更改而直接将 P_i 引用到指数中。根据 LPTF 原则和最小灵活度优先原则的特点, 生产时间长或者相邻生产时间小的工种应该被安排在安排表的开头或者结尾。也就是说, 如果一个工种不仅生产时间最长, 同时灵活度又最小, 那么它就该被安排在安排表的开始或者末尾。因此, 如果一个工种有比较小的 APTI(J_i), 它应该被安排在开头。

4.2. 构造性启发式算法

任何能够产生可行解的方法都能满足所提元启发式的第一部分, 但一个好的初始解能够减少计算时间。基于问题的特点, 这里提出一个简单的构造性启发式算法, 以得到一个好的解。这个启发式的主要观点是把单一机器安排表分截成两台相同并行机器的安排表, 关键的是截取单一机器安排表的哪一部分。我们将这个简单启发式称为“截一并二”(Cut One in Two, COIT), 其详细步骤如下:

1) 计算每个工种的 APTI, 选择 APTI 最小的工种放在首位。

2) 与前面的工种相比, 选择准备时间最短的工种放在第二位。若有准备时间最短的工种有多个, 选择 APTI 最小的那个工种。重复此操作直到所有的工种被分配完。这样就得到了单一机器从属序列时段安排表。计算该单一机器安排表的完成时间 C_s 。

确定截点, 以便截一并二, 且截点之前的工种分配在 M_1 上, 截点之后的分配在 M_2 上。此时有两种可能。第一, 如果点 $C_s/2$ 被包含在安排表的某个工种里, 则不小于 $C_s/2$ 的第一项工种可记为 J_{cut} , 其对应的执行处理时间为 C_i 。截一并二有两个潜在的截取点, 即 J_{cut} 对应的 C_i 或小于 $C_s/2$ 的最大的 C_i 。计算着两种截取方案的完成时间, 取较小的那个, 程序结束。第二, 如果 $C_s/2$ 被包含在安排表里两个相邻工种的某个准备时间中, 则直接在点 $C_s/2$ 进行截取, 然后计算 C_{max} , 程序结束。

根据案例工厂的数据, 即表 2, 执行 COIT 启发式算法如下。1) 由于 $\text{APTI}(J_9) = 0.3155$ 最小, 故 J_9 被分配的第一个工种; 2) 工种 J_5, J_6 与 J_9 有相同的准备时间 60 分钟, 但由于 $\text{APTI}(J_5) = 0.4074 < \text{APTI}(J_6) = 0.7515$, 故 J_5 为第二个被分配的工种。重复这个过程, 得到一份单一机器安排表, 即

$$(J_9, J_5, J_6, J_2, J_8, J_7, J_3, J_{11}, J_4, J_1, J_{10}) \quad (4.2)$$

其完成时间为 $C_s = 6521$ 。3) 由于安排表的中点 $C_s/2$ 落在 J_3 上, 故 $J_{\text{cut}} = J_3$ 。此时安排表截取为

$$M_1 = (J_9, J_5, J_6, J_2, J_8, J_7, J_3), M_2 = (J_{11}, J_4, J_1, J_{10}) \quad (4.3)$$

对应 $C_{\text{max}} = 3356$, 结束算法。不难看出, 这个结果要由于安排表(2.2)的 $C = 3367$ 。

4.3. 变邻域搜索算法

变邻域搜索算法(variable neighborhood search Algorithm, VNSA)是一种元启发式算法。VNSA 与多数局部搜索启发式不同之处在于, 在它的构造里, 至少有两个邻域。受 Hansen 等人[14]的启发, 在此我们针对所研究问题的特点, 对 VNSA 采用如下的邻域结构: 1) 工种插入: 任选一个工种, 随意把它插入安排表的任一位置; 2) 工种交换: 随意选取两个工种, 在安排表里把它们的位置调换。VNSA 构造的步骤说明如下:

(1) 执行 COIT 启发式, 找到初始解 x 。

(2) 重复以下步骤直到 I 迭代: a) 在 x 的邻域里, 利用插入工种随机产生可行解 x' ; b) 针对 x' , 应用工种交换 R 次, 选择最优解作为新的解 x'' ; c) 若 x'' 优于原有的结果, 则令 $x = x''$, 返回步骤(1); 否

则，返回步骤(2)。

依旧以表 2 为例。以安排表(4.3)与完成时间 $C_{\max} = 3356$ 作为初始解，当执行 VNS 的步骤(2)时，工种插入必须被单一机器安排表(4.2)采用，完成工作插入后，执行 COIT 启发式步骤 3，截取新的单一机器安排表，计算新的并行机器安排表的 C_{\max} 作为解 x' ；继续执行 VNS 的步骤，得到最终的安排表

$$M_1 = (J_9, J_{10}, J_3, J_7, J_1), M_2 = (J_6, J_5, J_4, J_{11}, J_8, J_2) \quad (4.4)$$

且对应的 $C_{\max} = 3255$ 。显然，这一结果不但明显由于 PVCL 厂现有的安排表(2.2)，也由于启发式算法所得的安排表(4.3)。在这个例子中， C_{\max} 的改进比例大约为 3.44%，但在削减两台机器总的准备时间方面，执行 VNS 相对减少了 165 分钟(当前工厂使用的方法需 630 分钟，VNS 需 465 分钟)。也就是说，总体准备时间的改进比例大约为 35.48%。

5. 计算结果评估

为了评估本文提出的 VNS 的性能，我们进行大量的计算实验。用 C++ 分别对当前工厂使用的排序方法，COIT 启发式和 VNS 元启发式进行编程。所有的数据都随机地产生于离散的一致分布，但分布参数不同。所有测试实例的处理时间都产生于离散一致分布 $U[1,100]$ 。对于工种数量 n 分别取值为 102,550,100 的情况，都有两种属性总数假设，即设 $|A| = 5$ 或 10 种。对于每组 $(n, |A|)$ ，属性准备时间 A_k 产生于三个离散一致分布，即 $U[1,25]$, $U[1,50]$ 和 $U[25,50]$ 。另外，对于每组 $(n, |A|, A_k)$ ，属性的层次总数设为 AL_k 产生两个离散一致分布，即 $U[2,5]$ 和 $U[5,10]$ 。因此，对于每个数量 n ，都有 12 种不同的组合，每组分别由 VNS、MIP 模型和 CM 法同环境下运行 10 次，然后对解求平均值。在我们提出的 VNS 实验里，设每个工种插入包含 $R = 100$ 次工种替换，迭代 $I = 100,000$ 。

我们进行三组实验：第一组实验是把 VNS 方法和 MIP 模型进行比较，针对于 $n = 10$ 即 10 个工种的问题，来评估 VNS 的效力；第二组实验是比较案例工厂当前使用的方法(以下简称为 CM 法)与 COIT 启发式，以展示 COIT 的性能；第三组实验是执行 VNS 对 COIT 的改进结果进行评估。

针对于第一组实验，测试实例取自于 $n = 10$ 个工种。针对每个测试实例，VNS 运行 10 次，得到最大值和最小值，其偏差 Dev 定义为 VNS 的最小解减去 MIP 模型的解，可利用程序自动计算。观察计算结果可知，由 12 种组合产生的 120 个测试实例的偏差 Dev 都等于 0。这表明当工种数目不多时，VNS 能够找到最优解。而且，大多数实例所得到的结果中，最大值与最小值相等。几乎所有的实例都能通过 MIP 模型在合理的时间范围之内解出，除其中之一花费了 545 秒外，其他都不超过 116 秒，且最少的仅仅使用了 1.72 秒。此外，我们注意到，10 种属性的情况，测试实例的平均运行时间(39.57 秒)要多于 5 种属性时的平均时间(24.35 秒)。可见，属性种类越多，问题越复杂，要很好地解决也越困难。

针对于第二组实验，将 COIT 启发式算法与 CM 法进行比较，以调查 COIT 的效力和效率。在这个实验中，工种数量 n 分别取值为 102,550,100。优化百分比定义为

$$PI = \frac{C_{\max}(\text{CM}) - C_{\max}(\text{COIT})}{C_{\max}(\text{COIT})} \cdot 100\%$$

观察计算结果可知，所有的优化百分比 PI 都是正的，故 COIT 启发式(在效益上)超过了当前工厂使用的方法。关于计算时间，由于 COIT 启发式算法与 CM 法都很简单，它们几乎不花费什么时间。此外，随着工种数目 n 的增加，优化百分比也大幅度地由 13.8% 提升到 51.09%，总体的平均改进比例为 35.23%。我们注意到， $|A| = 5$ 的测试实例得到的 PI 值稍微比 $|A| = 10$ 的要大些，这是因为有多种属性的实例解决起来更难、更复杂。从总体上看，属性准备时间 A_k 产生于分布 $U[1,50]$ 所得到的 PI 值，要显著性地大于产生于分布 $U[1,25]$ 和 $U[25,50]$ 所得到的 PI 值，这意味着属性准备时间分布更宽的实例能得到更好的改进。

对于属性层次的数目, AL_k 产生于 $U[2,5]$ 时的 PI 值要显著性地大于产生于 $U[5,10]$ 的 PI 值, 这主要是因为相比于属性层次多的实例, 属性层次少的更有可能得到更好的改进。在这个实验中, 如果一个例子的属性种类很少, 属性准备时间排列宽松并且属性层次很少, 那么 COIT 启发式算法将会得到很大的改进。如当 $n=100$, $|A|=5$, A_k 产生于分布 $U[1,50]$ 且 AL_k 产生于 $U[2,5]$ 时, PI 值为 66.85%, 这在所有组合里是最大的优化百分比。概括起来说, COIT 启发式算法比 CM 法更有效力和效率。

针对于第三组实验, 将我们所提出的 VNS 与 COIT 启发式算法进行比较, 以便评估初始结果的改善。在这个实验中, 工种数量 n 分别取值为 102, 550, 100。在每个测试实例中, VNS 运行 10 次得到一个平均结果值。其改进比例(PL)计算方法如下:

$$PI' = \frac{C_{\max}(\text{COIT}) - C_{\max}(\text{VNS})}{C_{\max}(\text{VNS})} \cdot 100\%$$

若 IP' 的值大于 0, 就表示 VNS 对 COIT 启发式得到的初始结果进行了改进。观察计算结果可知, 对于所有的测试实例, 随着工种数量 n 的增加, IP' 值明显地降低了; 当 $n=10$ 时, VNS 能做出最大的改进, 约为 7%, 当 $n=25$ 时, 平均改进比例小于 0.2%; 当工种数量更大时, 如 $n \geq 50$, 就几乎没有任何改进(但 $n=100$ 时, 其中有且只有一个实例 IP' 的值达到 5.61%, 这个例外并非是由于计算错误所致, 故耐人寻味)。总体而言, 工种数量 n 较小时, COIT 启发式很难得到较好的结果, 这是因为工种数量小导致计算范围也小, 所以这时候可以采用 VNS; 但随着工种数量的增加, 计算范围也在变大, 因此, 较长的处理时间跟计算范围的关系不是很明显, 这时 COIT 启发式得到的结果本身就比较理想, 以至于 VNS 都无法做出更大的改进。此外, 随着工种数量的增加, VNS 的计算时间也在稳步而小幅度地增加。虽然 COIT 不怎么花费时间, 但 VNS 也用极短的时间来改进结果。甚至在 $n=10$ 的例子中, VNS 也只用了少于 0.6 秒的计算时间, 对于所有的测试例子, VNS 平均用了 0.3 秒。

基金项目

本文受资助与国家自然科学基金(No. 11271069)。

参考文献

- [1] Abreu, C.F., May, J.H., Spangler, W.E., et al. (2008) Conflict Identification and Reconciliation in a Collaborative Manufacturing Scheduling Task. *International Journal of Information Technology & Decision Making*, 7, 147-174. <https://doi.org/10.1142/S0219622008002855>
- [2] Chen, W.J. (2009) Scheduling with Dependent Setups and Maintenance in a Textile Company. *Computers & Industrial Engineering*, 57, 867-873. <https://doi.org/10.1016/j.cie.2009.03.001>
- [3] Lee, C.H., Liao, C.J. and Chao, C.W. (2012) Scheduling with Multi-Attribute Setup Times. *Computers & Industrial Engineering*, 63, 494-502. <https://doi.org/10.1016/j.cie.2012.04.012>
- [4] Logendran, R., McDonell, B. and Smucker, B. (2007) Scheduling Unrelated Parallel Machines with Sequence-Dependent Setups. *Computers & Operations Research*, 34, 3420-3438. <https://doi.org/10.1016/j.cor.2006.02.006>
- [5] Chen, J.F. (2006) Minimization of Maximum Tardiness on Unrelated Parallel Machines with Process Restrictions and Setups. *The International Journal of Advanced Manufacturing Technology*, 29, 557-563. <https://doi.org/10.1007/BF02729109>
- [6] Abdekhodae, A.H., Wirth, A. and Gan, H.S. (2006) Scheduling Two Parallel Machines with a Single Server: The General Case. *Computers & Operations Research*, 33, 994-1009. <https://doi.org/10.1016/j.cor.2004.08.013>
- [7] Chan, F.T.S. and Choy, K.L. (2011) A Genetic Algorithm-Based Scheduler for Multiproduct Parallel Machine Sheet Metal Job Shop. *Expert Systems with Applications*, 38, 8703-8715. <https://doi.org/10.1016/j.eswa.2011.01.078>
- [8] Chuang, M.C., Liao, C.J. and Chao, C.W. (2010) Parallel Machine Scheduling with Preference of Machines. *International Journal of Production Research*, 48, 4139-4152. <https://doi.org/10.1080/00207540902991674>
- [9] Sun, K. and Li, H. (2010) Scheduling Problems with Multiple Maintenance Activities and Non-Preemptive Jobs on

- Two Identical Parallel Machines. *International Journal of Production Economics*, **124**, 151-158.
<https://doi.org/10.1016/j.ijpe.2009.10.018>
- [10] Pinedo, M. (2012) *Scheduling: Theory, Algorithms, and Systems*. 3rd Edition, Springer, Berlin.
<https://doi.org/10.1007/978-1-4614-2361-4>
- [11] Liao, C.J., Shyu, C.C. and Tseng, C.T. (2009) A Least Flexibility First Heuristic to Coordinate Setups in a Two- Or Three-Stage Supply Chain. *International Journal of Production Economics*, **117**, 127-135.
<https://doi.org/10.1016/j.ijpe.2008.10.002>
- [12] Gravel, M., Price, W.L. and Gagné, C. (2002) Scheduling Continuous Casting of Aluminum Using a Multiple Objective Ant Colony Optimization Metaheuristic. *European Journal of Operational Research*, **143**, 218-229.
[https://doi.org/10.1016/S0377-2217\(01\)00329-0](https://doi.org/10.1016/S0377-2217(01)00329-0)
- [13] Imran, A., Salhi, S. and Wassan, N.A. (2009) A Variable Neighborhood-Based Heuristic for the Heterogeneous Fleet Vehicle Routing Problem. *European Journal of Operational Research*, **197**, 509-518.
<https://doi.org/10.1016/j.ejor.2008.07.022>
- [14] Hansen, P., Mladenović, N. and Pérez, J.A.M. (2010) Variable Neighbourhood Search: Methods and Applications. *Annals of Operations Research*, **175**, 367-407. <https://doi.org/10.1007/s10479-009-0657-6>

知网检索的两种方式:

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>
下拉列表框选择: [ISSN], 输入期刊 ISSN: 2324-7991, 即可查询
2. 打开知网首页 <http://cnki.net/>
左侧“国际文献总库”进入, 输入文章标题, 即可查询

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: aam@hanspub.org