

基于维修时间窗的柔性作业车间调度优化研究

朱传军¹ 宋文家² 张超勇² 曹 静¹ 朱孟周³

1.湖北工业大学,武汉,430068

2.华中科技大学数字制造装备与技术国家重点实验室,武汉,430074

3.江苏省电力公司电力科学研究院,南京,211103

摘要:针对柔性作业车间调度和预防性维护的单目标集成优化问题,以最大完工时间为优化指标,建立了基于维修时间窗的集成优化模型,设计了混合“教与学”优化(HTLBO)算法求解该模型。提出一种“基于工序加工时间最短”的机器序列初始化策略,对部分初始种群进行初始优化,以提高部分初始解的质量,使得算法能够以较短的时间收敛。对文献中柔性作业车间调度的基准问题进行求解并比较其计算结果,初步证明该混合算法的可行性;针对集成维修时间窗的柔性作业车间调度优化模型,借鉴文献中的数据生成实例进行求解,并与其他算法进行比较,证明该混合算法的有效性。

关键词:维修时间窗;柔性作业车间调度问题;“教与学”优化;模拟退火

中图分类号:TP18

DOI:10.3969/j.issn.1004-132X.2016.10.010

Research on Optimization of FJSP Based on Maintenance Time Window

Zhu Chuanjun¹ Song Wenjia² Zhang Chaoyong² Cao Jing¹ Zhu Mengzhou³

1.Hubei University of Technology, Wuhan, 430068

2.State Key Laboratory of Digital Manufacturing Equipment & Technology,

Huazhong University of Science and Technology, Wuhan, 430074

3.Electric Power Research Institute, Jiangsu Electric Power Company, Nanjing, 211103

Abstract: An optimization model was proposed to minimize the makespan in FJSP based on maintenance time window, which integrated the preventive maintenance. A hybrid algorithm was presented to deal with the model based on TLBO. A new machine sequence initializing strategy was investigated to optimize partial initial population based on “shortest processing time for operation”, which might improve the quality of the initial solutions, so the computing process might convergence in a relatively short time. By calculating and comparing the results of FJSP benchmark problems with other literatures, the feasibility of the hybrid algorithm was proved on the whole. In view of flexible job-shop schedule optimization model which integrated maintenance time window, the data from the literatures was modified for the model and calculated by the hybrid algorithm. By comparing the results with other algorithms, the effectiveness of the hybrid TLBO algorithm was validated.

Key words: maintenance time window; flexible job shop scheduling problem(FJSP); teaching-learning-based optimization(TLBO); simulated annealing

0 引言

为消除实际生产中的设备失效和生产计划外中断等现象,需要定期或不定期地对设备进行预防性维护(preventive maintenance, PM)。按照设备维护时间可将预防性维护分为周期性维护和非周期性维护两类。一般来说,设备维护不可能在正常作业时进行,必须是作业结束后或开始前进行,所以周期性维护相对来说很难满足实际生产需要。目前关于周期性维护的研究文献相对较

少。Naderi等^[1]分析了柔性流水线车间的周期性预防性维护问题,采用基于遗传算法和人工免疫系统的两种启发式方法优化最大完工时间(makespan)。设备的非周期性预防性维护计划受到国内外学者的广泛关注。李林等^[2]建立了一种面向租赁设备的顺序预防维护策略,通过最小化期望总成本率获得了租赁设备的优化租赁期限及维护计划。夏唐斌等^[3]建立了一种设备层的单设备动态预防性维护的多目标决策模型。Ni等^[4]研究了生产过程中不影响产量的预防性维护机会,开发了一个识别维护机会的预测模型。非周期性预防性维护会影响实际生产计划,增加作业车间调度的难度,因此考虑设备非周期预防性维护的车间调度问题的研究也逐渐引起了人们的

收稿日期:2015-07-13

基金项目:国家自然科学基金国际(地区)合作与交流项目(51561125002);国家自然科学基金资助项目(51275190, 51575211);中央高校基本科研业务费专项资金资助项目(2014TS038)

关注。Fitouhi 等^[5]研究了针对单机非周期性预防维护的生产计划;Allaoui 等^[6]研究了双机柔性车间中带预防性维护的最小完工时间优化问题,假设条件为其中一台设备在调度开始的 T 个周期内必须完成一次预防性维护。针对柔性作业车间调度和预防性维护的集成优化问题,本文以最大完工时间为优化指标,在周期性维护的基础上建立了基于维护时间窗的集成优化模型,并设计了一种混合“教与学”优化(hybrid teaching-learning-based optimization, HTLBO)算法,对该问题进行求解。

1 考虑维修时间窗的柔性作业车间调度优化模型

1.1 问题描述

柔性作业车间调度问题(FJSP)可描述如下: n 个工件在 m 台机器上加工,每个工件分为 K 道工序,每道工序可以在若干台机器上加工,并且必须按一些可行的工艺次序进行加工;每台机器可以加工工件的若干工序,并且在不同的机器上加工的工序集可以不同。调度的目标是将工件合理地安排到各机器,使系统的某些性能指标达到最优。

FJSP 包括两个子问题:确定各个工件的加工机器和确定各个机器上各工件的加工先后顺序。调度的优化指标是最大完工时间,即所有工件都被加工完成的时间 $C_{\max} = \max_{1 \leq i \leq n} (C_i)$,其中 C_i 表示工件 J_i 的完工时间。

为了便于求解,本文假设以下条件:① 每台机床一次只能加工一个工件;② 机器的准备时间和工序间的转移时间可忽略不计;③ 所有工件均不包含可被打断的工序;④ 不同工件前后次序没有硬性要求;⑤ 分属不同工件的工序,其前后次序没有硬性要求;⑥ 对于隶属同一工件的所有工序,必须依照既定的顺序加工。

对于集成优化问题,假设每台机器在调度过程中均有固定的一个或者多个维护时间窗,所有机器均要在时间窗内进行预防性维护。在生产过程中,如果要进行设备的维护,必然要停止生产活动,这就造成了生产调度和设备维护的冲突。本文在柔性作业车间调度中安排设备维护时借鉴 Li 等^[7]的研究,提出一种“向前偏移”的预防性维护方法,如图 1 所示,PM 表示预防性维护, O_{ij} 表示工件 i 的第 j 道工序, t'_i 表示机器 i 的预防性维护时段的末尾时刻。具体步骤如下:

(1) 先按照传统柔性作业车间调度进行决策,不包含预防性维护,只包含工序的最初调度。

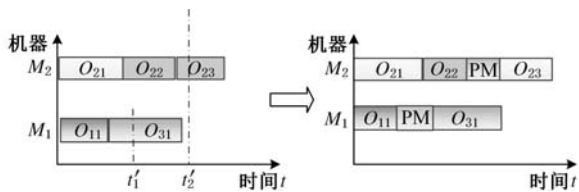


图 1 向前偏移的预防性维护方法描述图

(2) 将所有设备的维护安排在各个机器维护时间窗的末尾。

(3) 如果某一台机器预防性维护的时间段和其他工序的加工时间不冲突,则将该机器的维护和加工环节合并调度,否则执行步骤(4)。

(4) 如果某台机器的预防性维护时间段和某个工序加工的时间段发生冲突,则将维护环节尽量提前到前一道工序末尾或者机器开始加工的时刻,然后安排冲突的加工环节和之后的加工环节。

当然,预防性维护以后工件加工顺序如果改变,有可能会减小最大完工时间,但这些改变会在算法中的“教学”、“自学”及模拟退火环节中实现,因为这几个环节都对工序的排序情况进行变更。

1.2 数学模型

对设备的预防性维护和生产调度进行有机集成,把设备的维护考虑在作业计划之中,以最小化最大完工时间为目标,机器的预防性维护尽量安排在既定的时间窗内,制定设备维护和各个工序的生产计划。建立的数学模型如下:

$$\min f = \max_{j,k} (en_{jk}) \tag{1}$$

s.t.

$$be_{jk} + \sum_i X_{ijk} t_{ijk} = en_{jk} \quad \forall j, k \tag{2}$$

$$en_{jk} \leq be_{j(k+1)} \quad \forall j, k < K \tag{3}$$

$$\sum_{i=1}^{M_{jk}} X_{ijk} = 1 \quad \forall j, k \tag{4}$$

$$be_{jk} + t_{ijk} \leq be_{hl} + L(1 - y_{ijkhl}) \quad \forall i, j, k, h, l, j \neq h \tag{5}$$

$$[(MC_i - t_{pi} - en_{jk})X_{ijk} \geq 0] \vee [(en_{jk} - MC_i - t_{ijk})X_{ijk} \geq 0] \quad \forall i, j, k \tag{6}$$

$$\omega^{bi} + t_{pi} \leq MC_i \leq \omega^{ei} \quad \forall i \tag{7}$$

$$y_{ijkhl} \leq X_{ijk}, y_{ijkhl} \leq X_{ihl} \quad \forall i, j, k, h, l \tag{8}$$

$$X_{ijk} = \begin{cases} 1 & \text{工件 } j \text{ 的 } k \text{ 工序在机器 } i \text{ 上加工, } i \in M_{jk} \\ 0 & \text{否则} \end{cases}$$

$$y_{ijkhl} = \begin{cases} 1 & \text{机器 } i \text{ 上工件 } j \text{ 的 } k \text{ 工序先于工件 } h \text{ 的 } l \text{ 工序} \\ 0 & \text{否则} \end{cases}$$

式中, M_{jk} 为工件 j 的第 k 工序可用的机器集合, $M_{jk} \subseteq \{1, 2, \dots, m\}$; t_{ijk} 为工件 j 的第 k 工序在机器 i 上的加工时间, $i \in M_{jk}$; be_{jk} 为工件 j 的第 k 工序开始的时间; en_{jk} 为工件 j 的 k 工序完工时间; t_{pi} 为机器 i 预防性维护所需时间; MC_i 为机器 i 的维护结束时刻; L 为一个足够大的

正数; ω^{bi} 和 ω^{ei} 分别为机器 i 的预防性维护时段的开始和结束时刻。

式(1)为模型的目标函数,即最大完工时间最小;式(2)和式(3)表示每个工件的加工工序的顺序约束;式(4)表示机器约束,即同一时刻,一台机器能且只能加工一种工件的一个工序;式(5)表示在特定的时刻,一台机器只能加工一种工件的一种工序;式(6)表示同一台机器上设备预防性维护和工序的加工不能存在冲突;式(7)导入维修时间窗约束;式(8)说明只有分到同一机器上的各工序才需要进行排序。

2 HTLBO 算法求解集成优化问题

2.1 初始化

对于该优化问题,初始化时解的整体质量直接影响算法的收敛速度以及最终最优解的质量,所以必须选择比较好的编码方式,力求使初始化时产生的解的整体质量比较高。

柔性作业车间调度需要将每台机器上的不同工件的工序分配到合理的机器上,因此柔性作业车间调度中 HTLBO 算法采用两条编码,一条是基于工序的编码,用于说明不同工件的不同工序的加工先后顺序;另一条是基于机器的编码,用来确定具体每个工件的每个工序在哪一台机器上加工。

(1) 工序序列的编码。在基于工序的编码中,每个数字代表一个工件,数字出现的次数等于该数字对应工序的个数,且第 k 次出现的一个数字代表该数字对应的工件的第 k 个工序,例如,编码 [1 2 2 1 3 1 2 3] 表示工件 1 有三个工序,工件 2 有三个工序,工件 3 拥有两个工序。HTLBO 算法对基于工序的编码采用随机交换的方法,将所有工件的工序按照顺序依次排列,如 [1 1 1 2 2 2 3 3], 然后随机交换工序位置,产生该解内基于工序编码的序列,如 [1 3 1 2 2 1 3 2]。

(2) 机器序列的编码。基于机器的工序中,将各个工件的工序按照顺序排列下来,然后每个位置上对应的机器就是对应工序所在的机器。例如,编码 [1 3 1 2 2 1 3 2] 表明工件 1 的三个工序分别在机器 1、3、1 上加工;工件 2 的三个工序在机器 2、2、1 上加工;工件 3 的两个工序在机器 3、2 上加工。

为每个工序分配机器时要考虑该工序在不同可加工机器上的加工时间,最好选择最小值对应的机器来安排,同时需要兼顾考虑每台机器已经分配的负载情况,不能使某台机器的负载过大。

为了提高初始解的质量,借鉴 Kacem 等^[8]提出的利用时间表的分配方法 AL(approach by Localization) 和 Pezzella 等^[9]改进的分配规则来进行编码,使得编码的鲁棒性比较强,进而得到质量比较高的初始解。Pezzella 等^[9]改进的遗传算法编码是在 Kacem 等^[8]分配方法的基础上增加随机交换工件位置或者机器位置,以及在时间表中优先安排拥有全局最小加工时间的工序及机器,但是他们都没有考虑不同工件以及同一工件的不同工序的加工顺序。

本文在 Kacem 等^[8]分配方法的基础上对机器序列的初始化进一步改进,提出“基于工序加工时间最短”的机器序列初始化策略,即将加工时间的表格按照已经确定的工序排序从上到下对每行重新排序。具体过程如下:首先分配排在第一位的工序,选择加工该工序时间最短的某台机器分配给该工序;然后将表中该机器对应的加工时间位于该行之后的所有值均增加该工序的加工时间,表示该机器负载已经增加;最后对剩下的每个工序都进行同样的操作,最终确定对应该工序编码的机器分配方案。如表 1 所示,对于基于工序的序列 [1 3 1 2 2 1 3 2], M_i 为第 i 台机器,首先将工序 O_{11} 安排到最小加工时间的 M_3 , 将 M_3 剩余可加工的工序时间增加 4, 再安排 O_{31} , 最终的安排如表 1 最右边三列所示,得到的基于机器编码的序列为 [3 1 2 3 1 3 2 1]。

表 1 为已排序的工序安排加工机器的过程

工序	加工时间			累计加工时间			机器分配		
	M_1	M_2	M_3	M_1	M_2	M_3	M_1	M_2	M_3
O_{11}	7	6	4				7	6	4
O_{31}	4	8	5	4	8	9	4	8	5
O_{12}	9	5	4	9	5	8	9	5	4
O_{21}	2	5	1	2	5	5	2	5	1
O_{22}	4	6	8	4	6	12	4	6	8
O_{13}	9	7	2	9	7	6	9	7	2
O_{32}	8	6	3	8	6	7	8	6	3
O_{23}	3	5	8	3	5	12	3	5	8

由两列编码就可以结合上述动态安排设备预防性维护的方法来进行预防性维护安排,在某台机器上安排某工序后,若再安排其下一道工序将使得预防性维护与该工序冲突,则在该工序之后进行预防性维护。解码后画出对应的甘特图,见图 2。

由于要保证初始解的多样性,本文算法中一半初始解的机器编码采用表 1 所示的方法,另一半初始解的机器编码采用在该工序可用机器集合中随机分配机器的编码方法。

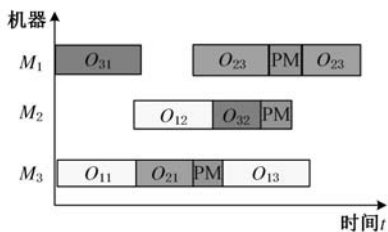


图 2 甘特图实例

2.2 “教与学”的过程

2.2.1 “教学”阶段

“教学”的过程和“学生”互相学习的过程借鉴遗传算法(GA)中交叉算子的思想,使得新解能从“教师”或者其他“学生”解的序列中“学习”到新知识。

由于要保证交叉变异以后得到的子代都是问题的可行解,因此一个解内的两条编码所需要的交叉变异方式是不相同的,表示工序次序的序列采用元素分集合的交叉方法,表示加工机器的序列采用多点交叉方法。交叉操作力求可以将较优秀序列的元素尽可能地保留在新得到的序列中。

(1) 工序序列串的交叉。工序序列的交叉过程是:首先把需要调度的工件集合 Q 任意分为非空子集 Q_1 和 Q_2 ,新解内的编码先继承“教师”集合 Q_1 内的工件对应的元素,然后将“学生”集合 Q_2 内的工件对应的元素分别填充到新解内的编码空缺的元素中,如图 3 所示,其中 $Q_1 = \{2, 3\}$ 。

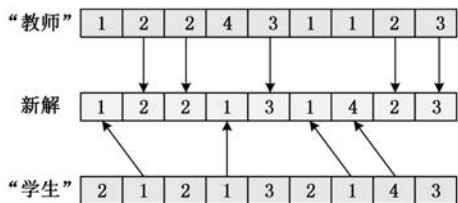


图 3 基于工序编码的交叉

(2) 基于机器编码的交叉采用多点交叉的方式,具体操作是:先随机产生一条和编码等长的 0-1 序列,将“教师”中与 0-1 序列中的 0 位置相同的所有元素复制到新解中,将“学生”中与 0-1 序列中的 1 位置相同的所有元素复制到新解中,如图 4 所示。

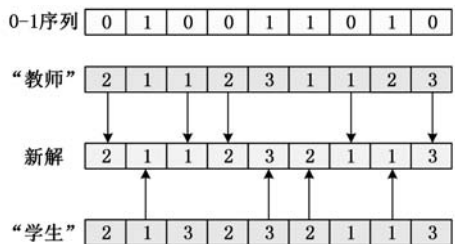


图 4 基于机器编码的交叉

2.2.2 “自学”阶段

算法中除了“教与学”的过程外,还增加了学

生“自学”的过程,即通过对“学生”解中的两条序列进行一定的操作,看是否能得到质量更优的解。自学的过程借鉴遗传算法变异操作的思想,对已有的序列进行一定的扰动来计算确定得到的解是否为更优解。

(1) 工序顺序序列的变异。表示工序顺序的序列串采用 Insert 变异,即从编码串中任意选取某元素,然后随机调换到序列的其他位置,如图 5 所示。

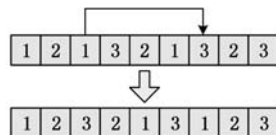


图 5 工序序列的变异

(2) 表示加工机器序列的变异。不同工件的不同工序可选择的机器各不相同,所以这部分编码的变异采用随机选取某个工件的某个工序的方式,将该工序的加工机器随机替换成该工序可选择机器集合中的其他机器,如图 6 所示。

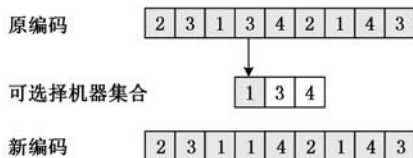


图 6 基于机器编码的变异

在教与学和自学阶段,算法均会产生新解。如果新解 S' 比原来的解 S 目标函数值更优,则一定用 S' 将 S 替换掉;如果 S' 不比 S 的结果更优,则引入一个选择因子 $c = rand(0,1)$,如果 $c \geq 0.5$ 则将 S' 保留在算法解集中,否则丢弃 S' 。这样可以使得到的更优解保留下来,选择性保留非更优解既可以保证算法中解的多样性,还可以避免算法中解的数量过于庞大,影响算法的空间复杂度。

2.3 模拟退火算法

基本 TLBO 算法求解一定时间后很可能会陷入局部最优,可以将邻域结构的局部搜索加入该算法中,以提高算法整体的搜索效率^[10-12]。本文将模拟退火(simulated annealing, SA)算法加入到 TLBO 算法中,在算法中“教与学”和“自学”之后,采用 SA 算法对所得的解进行局部搜索,搜索对象是原有解集和产生的所有解,这样会导致算法整体时间复杂度有所增加,但可以很大程度地提高解的质量。

SA 算法是一种根据热力学统计定律得出的邻域搜索技术,在过去求解组合问题时产生了良好的效果^[13]。标准的 SA 算法首先随机产生初始解,设定初始温度 T_0 、终止温度 T_f 、温度衰减率

α , 然后运用特定的邻域结构对已得到的解进行邻域搜索, 若得到的新解更优, 就用新解将旧解替换掉。随着算法的进行, 温度也以一定比例降低, 直到达到设定的终止温度 T_f 。

在 SA 算法环节中, 对于两条编码需要用不同的邻域结构进行局部搜索。采用两种基于工序编码的邻域结构, 分别是反向 (Inverse) 和交换 (Swap)。反向的邻域结构就是在编码中随机选择一段编码, 将所有元素逆转反向排序; 交换的邻域结构就是在编码中随机选择两个元素交换位置。两种邻域结构如图 7 所示。

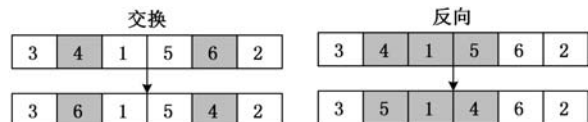


图 7 基于工序编码的邻域结构

对于已知解中基于机器编码的邻域结构, 采用本文算法提出的机器编码的变异方式进行变异。混合算法中模拟退火的步骤如下: ① 设定初始温度 T_0 、终止温度 T_f 、温度衰减率 α 的值; ② 选择已有解 S , 运用 SA 算法的邻域结构对 S 进行局部搜索得到 S' , 令 $\Delta S = f(S') - f(S)$; ③ 如果 $\Delta S < 0$ 则将 S 用 S' 替换掉, 否则令 $P = \exp(-\Delta S/T)$ 决定是否用 S' 替换 S ; ④ 使温度以衰减率 α 缓慢降低, 即 $T_{k+1} = T_k \alpha$, 其中 $\alpha \in (0, 1)$; ⑤ 重复步骤 ① ~ ④, 直至满足停止条件。

由于要保证算法空间和时间复杂度不至于过高, 所以模拟退火阶段得到的新解 S'' 只有比之前的解 S' 更优才会得以保留, 若 S'' 比之前的解 S' 更优则将 S' 替换掉。

2.4 算法执行流程

HTLBO 算法流程如图 8 所示, P 的值由混合算法中模拟退火的步骤 ③ 获取, 该算法具体步骤如下: ① 按照 2.1 节所述进行算法的初始化, 产生初始解集 P_t , 此时 $t \leftarrow 0$, 令解集规模为 N ; ② 在初始解集中选择 n 个目标值较好的解作为“教师”; ③ 按照 2.2 节所述对解集 P_t 进行“教与学”的过程和“自学”过程, 得到新的解集 Q_t ; ④ 运用 SA 算法对 Q_t 中的解进行搜索, 得到新的解集 Q_t' ; ⑤ 如果 $t < t_{\max}$ (最大迭代次数), 则 $t \leftarrow t + 1$, 返回步骤 ③; 否则算法终止; ⑥ 输出算法求得的最优解集的最优解。

3 实例验证与分析

为验证 HTLBO 算法的有效性, 首先对文献 [14] 中的 10 个 MK 算例进行测试, 均以最小化最

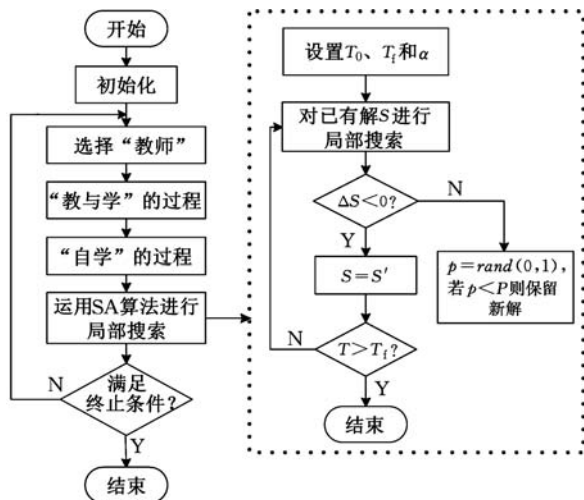


图 8 HTLBO 算法流程图

大完工时间为目标, 并与其他文献中的算法的测试结果进行对比。同时参考其他文献的数据生成三个实际模型的算例进行求解, 与基本 TLBO 算法和遗传算法求解的结果进行对比。

3.1 基准实例的测试

HTLBO 算法采用 C++ 语言编程, 计算机运行环境为 2.5 GHz Intel Core i5 多核 CPU 和 2 GB RAM, 算法的有关参数设置如下: 种群规模 $Popsize = 200$, 遗传代数 $Generation = 100$; 模拟退火环节中, 本文取初始温度 $T = 1000$, 温度衰减率 $\alpha = 0.8$, 终止温度 $T_f = 1$, 在温度 T 时, 每个已有解进行 5 次的扰动。

MK 实例的 HTLBO 算法测试结果与其他文献算法结果对比如表 2 所示, 其中, m 和 n 分别表示每个问题中机器和工件的数量, GA-Chen 表示 Chen 等 [15] 的测试结果, GA-Jia 表示 Jia 等 [16] 的测试结果, GA-Pezzella 表示 Pezzella 等 [9] 的测试结果, TSPCB 表示 Li 等 [17] 的测试结果, C 表示测试目标 makespan 的值; Div 为最优解相对偏差的值, 其计算公式为

$$Div = \frac{C_i - C}{C_i} \quad i = 1, 2, 3, 4 \quad (9)$$

式中, C_i 为对比算法求得的 makespan 的值; C 为 HTLBO 算法求得的 makespan 的值。

由表 2 可以看出, HTLBO 算法求解出了质量比较高的解 (达到或者超过其他算法求解出的 makespan 最优值), 在表中已用黑体突出显示。

从表 2 中数据可以看出, 对于前 6 个基准问题以及 MK08 问题, HTLBO 算法均求解出了已有文献中得出的最优解; 对于 MK10 问题, HTLBO 算法得到的解比其他 4 种算法的解更优; 只有 MK07 问题和 MK09 问题没有得到其他算法的最优解, 但相差不大。

表 2 10 个 MK 实例测试结果对比

实例	n	m	HTLBO C	GA - Chen		GA - Jia		GA - Pezzella		TSPCB	
				C ₁	Div(%)	C ₂	Div(%)	C ₃	Div(%)	C ₄	Div(%)
MK01	10	6	40	40	0.00	40	0.00	40	0.00	40	0.00
MK02	10	6	26	29	10.34	28	7.14	26	0.00	26	0.00
MK03	15	8	204	204	0.00	204	0.00	204	0.00	204	0.00
MK04	15	8	60	63	4.76	61	1.64	60	0.00	62	3.22
MK05	15	4	172	181	4.97	176	2.27	173	0.58	172	0.00
MK06	10	10	60	60	0.00	62	3.22	63	4.76	65	7.69
MK07	20	5	140	148	5.41	145	3.45	139	-0.72	140	0.00
MK08	20	10	523	523	0.00	523	0.00	523	0.00	523	0.00
MK09	20	10	310	308	-0.65	310	0.00	311	0.32	310	0.00
MK10	20	15	208	212	1.89	216	3.70	212	1.89	214	2.80

3.2 考虑维修时间窗的实例测试

对于具体实例的测试,传统的 FJSP 测试数据来自文献[14]中的 MK04、MK07 和 MK09 的三组数据,三个实例分别是 15×8、20×5 以及 20×10 的 FJSP 问题。对于三个问题中各台机器的预防性维护时间窗和预防性维护需要的时间 t_{pi} ,假定分别如表3~表5所示, PM_{mm} 表示第 m 台机器的 n 次维护。先用 HTLBO 算法获得三个结果,再分别用 GA 算法的解进行对比。

三种算法对三个算例分别进行求解得到的结果对比如表6所示。由表6可以看出,针对本文建立的单目标集成优化模型,HTLBO 算法对三个算例求得的结果均优于对比算法 GA 算法的结果,因此,本文提出的 HTLBO 算法求解基于固定维护时间窗的柔性作业车间调度和预防性维护的集成优化问题是有效的。其中,对于第一个 15×8 的算例,HTLBO 算法求解出的最优解的甘特图见图9,图中某些工序之后紧跟的“PM”小条即是预防性维护的环节。 J_m 表示第 m 个工件,工序的先后顺序由甘特图中的时间先后来决定。

表 3 15×8 问题的预防性维护时间数据

	PM_{11}	PM_{21}	PM_{31}	PM_{41}	PM_{51}	PM_{61}	PM_{71}	PM_{81}
τ^{bi}	21	13	15	34	22	20	23	19
τ^{ei}	34	29	25	42	28	31	35	26
t_{pi}	3	2	4	2	2	4	2	3

表 4 20×5 问题的预防性维护时间数据

	PM_{11}	PM_{21}	PM_{31}	PM_{41}	PM_{51}
τ^{bi}	20	21	30	20	32
τ^{ei}	34	27	40	31	39
t_{pi}	2	1	1	2	1

表 5 20×10 问题的预防性维护时间数据

	PM_{11}	PM_{21}	PM_{31}	PM_{32}	PM_{41}	PM_{51}
τ^{bi}	21	32	20	53	30	42
τ^{ei}	35	47	33	61	40	58
t_{pi}	1	1	1	2	3	2
	PM_{61}	PM_{71}	PM_{81}	PM_{82}	PM_{91}	$PM_{10,1}$
τ^{bi}	30	52	20	66	31	52
τ^{ei}	46	67	35	81	45	68
t_{pi}	1	1	1	1	1	1

表 6 三种具体实例的测试结果对比

15×8 问题		20×5 问题		20×10 问题	
HTLBO	GA	HTLBO	GA	HTLBO	GA
74	77	157	160	331	335

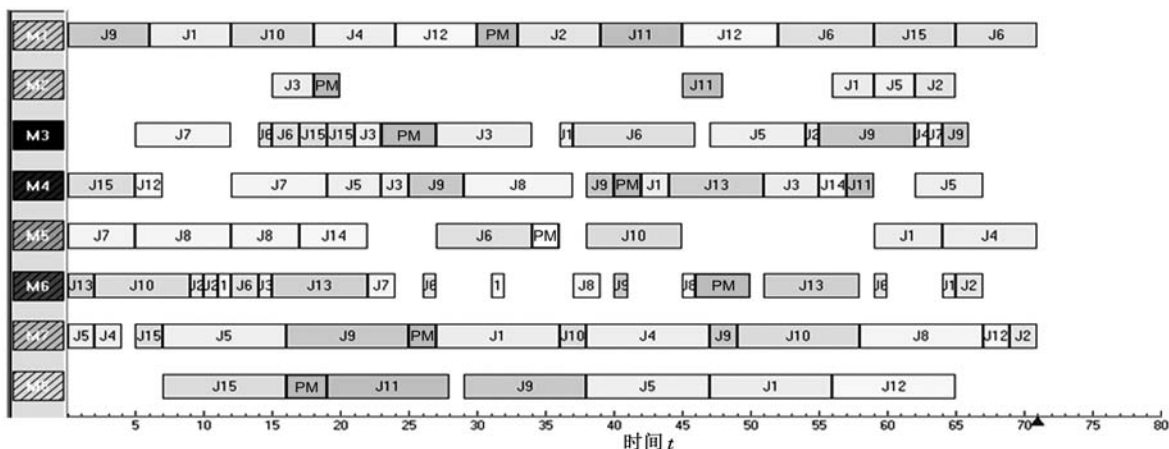


图 9 15×8 算例最优解的甘特图

4 结论

针对柔性作业车间和预防性维护的集成优化

问题,建立了基于机器维护时间窗的集成优化模型,在基本“教与学”优化(TLBO)算法中加入模拟退火的过程,提出了新颖的混合“教与学”优化

(HTLBO)算法。该算法结合了 TLBO 算法全局搜索的优势和模拟退火算法的局部搜索能力。将该算法用于求解 10 个 MK 基准测试实例,并与其他文献得到的结果进行对比,初步证明了该算法的有效性。选择其中三个实例,针对其特点分别生成三个符合本文模型的集成预防性维护的具体实例,运用 HTLBO 算法得到最优解,并与遗传算法得到的最优调度进行比较,验证了 HTLBO 算法在解决基于机器维护时间窗的柔性作业车间和预防性维护的集成优化问题的有效性。

参考文献:

[1] Naderi B, Zandieh M, Aminnayeri M. Incorporating Periodic Preventive Maintenance into Flexible Flow-shop Scheduling Problems[J]. Applied Soft Computing, 2011,11: 2094-2101

[2] 李林,金琳,周晓军. 基于冲击建模的租赁设备预防维护策略[J]. 计算机集成制造系统,2013,19(1): 114-118.
Li Lin, Jin Lin, Zhou Xiaojun. Preventive Maintenance Strategy Based on Shock Modeling for Leasing Equipment[J]. Computer Integrated Manufacturing Systems, 2013,19(1):114-118.

[3] 夏唐斌,奚立峰,周晓军,等. 串并联衰退系统的多目标预防性维护优化建模[J]. 计算机集成制造系统,2010, 16(4):783-788.
Xia Tangbin, Xi Lifeng, Zhou Xiaojun, et al. Optimization of Multi-objective Preventive Maintenance for Series-parallel Deteriorating Systems[J]. Computer Integrated Manufacturing Systems, 2010, 16(4):783-788.

[4] Ni Jun, Gu Xi, Jin Xiaoning. Preventive Maintenance Opportunities for Large Production Systems [J]. CIRP Annals — Manufacturing Technology, 2015,64: 447-450.

[5] Fitouhi M-C, Nourelfath M. Integrating Noncyclical Preventive Maintenance Scheduling and Production Planning for a Single Machine[J]. Int. J. Production Economics, 2012,136:344-351.

[6] Allaoui H, Lamouri S, Artiba A, et al. Simultaneously Scheduling n Jobs and the Preventive Maintenance on the Two-machine Flow Shop to Minimize the Makespan[J]. Int. J. Production Economics, 2008,112 : 161-167.

[7] Li J Q, Pan Q K. Chemical-reaction Optimization for Flexible Job-shop Scheduling Problems with Maintenance Activity[J]. Applied Soft Computing,

2012,12(9): 2896-2912.

[8] Kacem I, Hammadi S, Borne P. Approach By Localization and Multiobjective Evolutionary Optimization for Flexible Job-shop Scheduling Problems [J]. Applications and Reviews, IEEE Transactions on Systems, Man, and Cybernetics, Part C, 2002, 32(1): 1-13.

[9] Pezzella F, Morganti G, Ciaschetti G. A Genetic Algorithm for the Flexible Job-shop Scheduling Problem[J]. Computers & Operations Research, 2008,35(10): 3202-3212.

[10] Mladenovic' N, Hansen P. Variable Neighborhood Search[J]. Computers & Operations Research, 1997, 24(11):1097-1100.

[11] Battiti R, Tecchiolli G.The Reactive Tabu Search [J]. ORSA Journal on Computing, 1994, 6(2): 126-140.

[12] Eglese R. Simulated Annealing: a Tool for Operational Research[J]. European Journal of Operational Research, 1990, 46(3): 271-281.

[13] Low C. Simulated Annealing Heuristic for Flow Shop Scheduling Problems with Unrelated Parallel Machines[J]. Computers & Operations Research, 2005, 32(8): 2013-2025.

[14] Brandimarte P.Routing and Scheduling in a Flexible Job Shop by Tabu Search[J]. Annals of Operations Research, 1993, 41(3): 157-183.

[15] Chen H, Ihlow J, Lehmann C. A genetic Algorithm for Flexible Job-shop scheduling[C]//1999 IEEE International Conference on Robotics and Automation. Detroit: IEEE,1999:1120-1125.

[16] Jia H, Nee A, Fuh J, et al. A Modified Genetic Algorithm for Distributed Scheduling Problems [J]. Journal of Intelligent Manufacturing, 2003,14(3/4): 351-62.

[17] Li J Q, Pan Q K, Suganthan P, et al. A Hybrid Tabu Search Algorithm with an Efficient Neighborhood Structure for the Flexible Job Shop Scheduling Problem [J]. The International Journal of Advanced Manufacturing Technology, 2011,52(5/8): 683-697.

(编辑 苏卫国)

作者简介:朱传军,男,1971年生。湖北工业大学机械工程学院副教授、博士。主要研究方向为智能优化算法、决策分析。宋文家,男,1991年生。华中科技大学数字制造装备与技术国家重点实验室硕士研究生。张超勇,男,1972年生。华中科技大学机械科学与工程学院副教授、博士。曹静,女,1970年生。湖北工业大学机械工程学院讲师。朱孟周,男,1982年生。江苏省电力公司电力科学研究院工程师、博士。