

Research on Flexible Genetic Shop Scheduling with Mixed Genetic Simulated Annealing Considering Balanced Load

Duo Shao

School of Computer Science and Technology, Tianjin Polytechnic University, Tianjin
Email: 382300501@qq.com

Received: Jul. 6th, 2019; accepted: Jul. 19th, 2019; published: Jul. 26th, 2019

Abstract

In this paper, a hybrid genetic simulated annealing algorithm is proposed for flexible shop scheduling problem. The algorithm uses genetic algorithm to solve the problem of sorting and global search. It uses simulated annealing to perform local search. In the process of solving, the machine is allocated according to the load condition of the machine. The effectiveness of the algorithm is verified by an example test.

Keywords

Flexible Job-Shop Scheduling, GASA, Balanced Load

考虑平衡负荷的混合遗传模拟退火柔性车间调度度的研究

邵 铎

天津工业大学, 计算机科学与技术学院, 天津
Email: 382300501@qq.com

收稿日期: 2019年7月6日; 录用日期: 2019年7月19日; 发布日期: 2019年7月26日

摘 要

本文针对柔性车间调度问题, 提出一种混合遗传模拟退火算法, 该算法采用遗传算法解决排序和全局搜

索问题,使用模拟退火进行局部搜索,在求解过程中根据机器负载情况合理的对机器进行分配,并通过实例测试,验证了算法的有效性。

关键词

柔性作业车间调度,遗传模拟退火,平衡负荷

Copyright © 2019 by author(s) 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. 引言

柔性车间调度问题[1] (Flexible Job Shop Scheduling Problem, 简称 FJSSP)是 Bruker 和 Schlie 在 1990 年首次提出的。与经典车间作业调度相比较,柔性作业车间调度突破了资源唯一性的限制,增加了作业调度的灵活性,从而更适合智能化车间的生产要求。但是,柔性车间调度问题是一个更为复杂的 NP 问题,因此,研究 FJSSP 具有重要的应用价值和意义。

目前对柔性作业车间调度的主要研究方法为粒子群算法[2] [3]、遗传算法[4] [5]、蚁群算法[6]、邻域搜索算法[7]等算法,通过这些算法的迭代实现柔性作业车间调度方案的优化。黄学文[8]等人提出的基于 OR 子图和子路径的工艺路径调度算法通过遗传较好的实现了可行解;赵卫[9]在遗传算法种群更新过程中引入模拟退火机制,提高了种群更新迭代中多样性、加快了收敛、克服了早熟的缺陷,提高了调度的效率。这些学者研究车间调度算法及其优化都只是针对柔性车间中某一道工序指定一台加工设备而言,不能应用于一道工序指定多台加工设备的分配和调度问题。为了更好的解决此问题,陈成[10]等人使用蚁群遗传混合算法解决了机器分配及工序排序,能够有效的解决一道工序可由多台设备加工的分配和调度问题,但是使用轮盘赌法增加了局部最优值的不确定性。

因此,面对智能化车间调度系统,考虑到柔性车间中每台加工设备的性能、型号、用途等不尽相同,但每个加工设备的通用性强,加工工件的种类和工序的数量较多的前提,使用传统的一道工序必须由指定一台设备加工的调度方案已经不能够适应生产需求的因素,本文从两个方向对 FJSSP 进行设计和优化,1) 对车间调度本身的算法进行了负荷平衡的改进和优化,有效的解决了机器合理的选择与分配问题。2) 基于优化的混合遗传算法和模拟退火算法,利用遗传算法隐式并行处理能力和强鲁棒性的特点寻找全局最优,利用模拟退火算法在大规模的解空间中寻找局部最优,提高了车间调度的效率,并通过车间作业调度测试案例仿真验证算法的有效性。

2. 作业车间调度问题

2.1. 作业车间调度问题表述及其约束条件

在流水车间中,对于 n 个待加工工件进行加工,每个工件又具有一个或者多个加工工序,每个工件依照加工流程中工序顺序进行排产,考虑到柔性车间中所有的机器的性能,型号,用途等不尽相同,机器加工工件的种类、同一工件的工序、同一工序的时间也都不尽相同。因此考虑调度问题是要讲各个工序分配到各台机器上,并合理地安排工序的加工次序和时间,从而尽可能满足性能指标,这增加了问题的复杂性和求解难度,但更加符合实际的多样化生产环境。

FJSSP 可以描述为:

- 1) 工件集 $J = \{J_1, J_2, \dots, J_n\}$, J_i 为第 i 个工件, $1 \leq i \leq n$;
- 2) 机器集 $M = \{M_1, M_2, \dots, M_m\}$, M_j 为第 j 号机器, $1 \leq j \leq m$;
- 3) 工序集 $O = \{O_1, O_2, \dots, O_n\}$, O_i 为第 i 个工件的工序, $1 \leq i \leq n$;
- 4) 工序子集 $O_i = \{O_{i1}, O_{i2}, \dots, O_{im}\}$, O_{ik} 为第 i 个工件的第 k 道工序使用的机器号, $1 \leq k \leq m$; $O_{ik} = 1$ 表示工件 J_i 的第 k 道工序由机器号加工, $1 \leq k \leq m$;
- 5) 每道工序加工的使用时间 $T_i = \{T_{i1}, T_{i2}, \dots, T_{im}\}$, T_{ik} 为第 i 个工件的第 k 道工序所使用的机器号完成所消耗的时间, $1 \leq k \leq m$ 。

FJSSP 需要满足以下的约束条件:

- 1) 每个加工机器彼此互相独立, 不考虑机器故障带来的影响, 并且所有的机器都是从($t = 0$)时刻开始工作。
- 2) 所有的工件彼此互相独立, 任一工件的加工不会影响其他工件的加工, 所有工件均可在开始时刻加工。
- 3) 加工过程中不能突然改变工件工艺, 所有的工件的工序在排产以前就已经确定。
- 4) 所有机器的加工每件的时间由机器本身的加工效率和工艺决定和确定, 加工过程中不会对任何工件的任一工序的时间有所改变。
- 5) 每台设备每个时刻只能加工一个工件的一个工艺, 同一工艺不能在多台机器上加工, 一道工艺在某台加工设备上一旦开始加工, 直至该工艺加工完成, 中间不能中断。
- 6) 加工机器严格遵守工件工序的加工时间。

本文研究最小化最大完工时间调度目标的 FJSSP。只考虑工件在加工设备上的加工时间, 不考虑加工设备的调整时间以及工件在各个加工设备之间的运输时间。因此, 一个工件的完工时间等于该工件的各个工艺在相应加工设备上的加工时间和等待时间之和。

目标函数是最小化最大时间, 即:

$$C = \min \left(\max_{1 \leq j \leq n} \left(\max_{1 \leq k \leq m} e_j \right) \right) \quad (1)$$

其中的 e_j 表示为第 j 号机器的完工时间。

2.2. 平衡负荷的设计

由于每道工序都可以由一台或多台设备加工, 系统在为工件加工分配机器的时候往往并行机器需要合理分配加工任务, 均衡设备负载, 提高整体运作效率。通过计算每台机器负载均方差来反应当前机器的负载情况, 负载均方差越小, 该机器的负荷越小, 越应优先分配加工任务。

假设调度系统正在分配工件 J_i 的第 k 道工序, 参考工序集 O_i 可以确定由工序子集 O_{ik} 的 k 台机器来加工, 设 T_j 为第 j 台机器当前加工总时长; T_{total} 为所有机器加工总时长; \bar{T} 为每台设备的平均加工时长; Nlb 表示负载因子, Nlb 越小表明该机器优先分配第 i 工件的第 k 道工序越合理。

构建平衡负荷数学模型如下:

$$T_{\text{total}} = \sum_{j=1}^m T_j \quad (2)$$

$$\bar{T} = \frac{T_{\text{total}}}{M} \quad (3)$$

$$Nlb = \sqrt{\sum_{j=1}^k (T_j + T_{ij} - \bar{T}) / M} \quad (4)$$

3. 基于遗传和模拟退火的车间调度算法

遗传算法(Genetic Algorithm, GA)的主要思想基于达尔文进化论和孟德尔的遗传学,它结合了达尔文的适者生存和随机交换理论,是一种自然进化系统的计算模型,也是一种通用求解优化问题的搜索方法。遗传算法后期适应度趋向一致,优秀个体在产生后代时,优势不明显使得整个种群进化停滞。因此有必要对适应度进行适当的拉伸,当在温度较高的时候,适应度详尽的个体产生后代的概率相近,而温度下降的时候拉伸作用加强,使得适应度相近的个体适应度差异放大,从而使优秀个体优秀更显著。既有效的克服了遗传算法的早熟现象,同时又能有效的快速收敛得到全局最优解。

模拟退火算法(Simulated Annealing Algorithm, SA)是1982年由Kirkpatrick等人提出的一种模拟金属退火机理而建立的随机优化方法。模拟退火算法是局部搜索算法的扩展,它不同于局部搜索算法的支出是以一定的概率选择邻域中费用值大的状态,适用于解决大规模组合优化问题。

3.1. 编码

本文采用工件的整数排列的编码方法,即以工件为基本变量,工件号的排序作为编码,代表着染色体的基因组成。一个工件可以在染色体中出现多次,表示该工件的多个工序。比如有5个工件的染色体编码:“11223443355”其中1、2、3、4、5为工件号,分别重复出现了2、2、3、2、2次,这些也说明了其工序数量分别是2、2、3、2、2道,且每次出现的顺序代表的是工序顺序,编码中的9位的“3”是第二次出现,所以第9位代表的是工件3的第二道工序。该编码的优点在于工件工序变化不定的情况能够灵活适用。

3.2. 适应度函数设计

适应度函数决定了基因序列的好坏程度,适应度函数一般是目标函数的倒数,即优化的目标是适应度函数值尽量选择最小化最大时间最小的染色体序列,适应度函数值越大染色体越优质,反之亦反。目标函数公式为:

$$F = \frac{1}{P_n} \quad (5)$$

3.3. 选择操作

选择操作即从旧群体中以一定概率选择个体到新群体中,个体被选中的概率跟适应度的值有关,个体适应度值越大,被选中的概率越大。操作步骤如下:

1) 首先确定种群在下一代中的生存数目:

$$N^* = \sum_{i=1}^N \left\lfloor N * F_i / \sum_{i=1}^N F_i \right\rfloor \quad (i=1,2,\dots,N) \quad (6)$$

N 为种群的规模, F_i 为第*i*个染色体的适应度值。

2) 根据二元锦标赛方式选择较好的个体,即每次从种群中随机选出两个个体进行对比,将适应度较高的个体保存到下一代中,重复这个过程直至下一代新种群的数量达到预期数量 N^* 。

$$C_i^* = \begin{cases} C_i, & F_i > F_{i+1} \\ C_{i+1}, & F_i < F_{i+1} \end{cases} \quad (7)$$

比较 C_i 和 C_{i+1} 两个个体的适应度值,保留适应度值大的个体进入下一代的种群中。

3.4. 交叉操作

种群的交叉操作是为了得到有益的新基因染色体。通过交叉操作让染色体有规律的基因进行交换从

而得到两个新的染色体。这个过程要确保父代中的优良信息能够遗传到子代，尽量少的将不必要的遗传信息传给子代。本文采用一种工序编码的 PBX 算法(Position-based Crossover)的改进算法具体流程如下：

- 1) 在种群中随机选取两个 P_1 和 P_2 染色体作为父代染色体，并初始化基因个数与父代个数相同的 C_1 和 C_2 ，作为下一代个体。
 - 2) 首先随机划分工件集 $\{1,2,\dots,n\}$ 为一个非空子集 set_1, set_2 。 $set_1 \cap set_2 = \emptyset$ 。
 - 3) 将 P_1 中存在的与 set_1 集合中相同的基因复制到 C_1 中，复制到 C_1 中的基因位置也与在 P_1 位置对应相同。
 - 4) 将 P_2 中不存在于 set_1 集合中的元素按顺序复制到 C_1 的空缺位置中，如图 1。
- 交叉概率的自适应调整，其公式为：

$$P_c = \begin{cases} 0.9 - 0.3 \frac{F' - F_{avg}}{F_{max} - F_{avg}}, & F' \geq F_{avg} \\ 0.9, & F' \leq F_{avg} \end{cases} \quad (8)$$

公式中： P_c 为交叉概率； F_{max} 是群体中最大的适应度。 F_{avg} 为每一代群体的平均适应度值， F' 为交叉后染色体的适应度值。

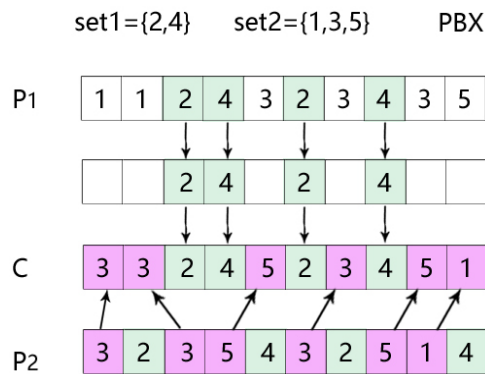


Figure 1. PBX cross graph
图 1. PBX 交叉图

3.5. 变异操作

变异操作是将个体染色体中的基因编码序列改变从而达到新的个体的操作。它能改善遗传算法的局部搜索能力，维持种群的多样性，防止“早熟”的发生。本文的变异方法使用随机选择位置置换的方法。随机选择两个变异的位置，互相交换即可，如图 2。

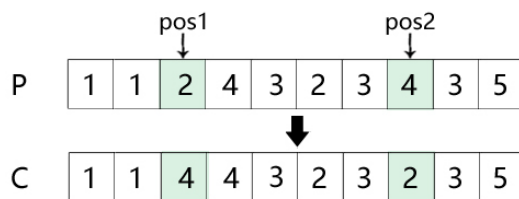


Figure 2. Gene variation graph
图 2. 基因变异图

变异操作使用自适应概率，公式为：

$$P_m = \begin{cases} 0.9 - 0.3 \frac{F' - F_{\text{avg}}}{F_{\text{max}} - F_{\text{avg}}}, & F' \geq F_{\text{avg}} \\ 0.9, & F' \leq F_{\text{avg}} \end{cases} \quad (9)$$

其中 F_{avg} 为每一代群体的平均适应度值, F' 为交叉后染色体的适应度值。

3.6. 模拟退火操作

本文中对模拟退火进行改进, 将模拟退火结合遗传算法的交叉、变异操作, 本算法既使收敛速度加快, 又要维护种群的多样性, 过程如图:

- 1) 初始化: 初始温度为 T (充分大), 终止温度 T_{min} , 初始化种群 P , 令当前状态为 P_i , 最优解 $P^* = P_i$, $t=0$, 设置循环计数器 k , 降温系数 d , L 表示 *Metropolis* 算法第次迭代时产生的变换个数;
- 2) 产生新状态 S' , 计算增量 $\Delta f' = f(P'_i) - f(P_i)$;
- 3) 判断 $\Delta C'$, 若 $\Delta C' > 0$, 以概率 $\exp(-\Delta f'/t)$ 接受 P' 为当前的状态, P' 被接受则令 $P' = P$, 令 $t = t + 1$; 否则接受 P' 为当前的状态, 计算增量 $\Delta C' = C(P'_i) - C(P_i)$ 若 $\Delta C' > 0$ 则 $P^* = P'$, 令 $t = 0$;
- 4) 判断是否 $t \geq L_k$, 当前的状态是否无变化, 如果是则结束循环;
- 5) 判断 k 是否终止, 如果终止则令 $k = k + 1$; 转到步骤 2, 否则结束;
- 6) 重复以上操作直到对种群中的所有个体都完成了搜索。

3.7. 算法流程

步骤 1: 初始化种群大小、初始温度、交叉概率等模拟退火和遗传算法中需要的各种参数。

步骤 2: 随机产生初始种群。

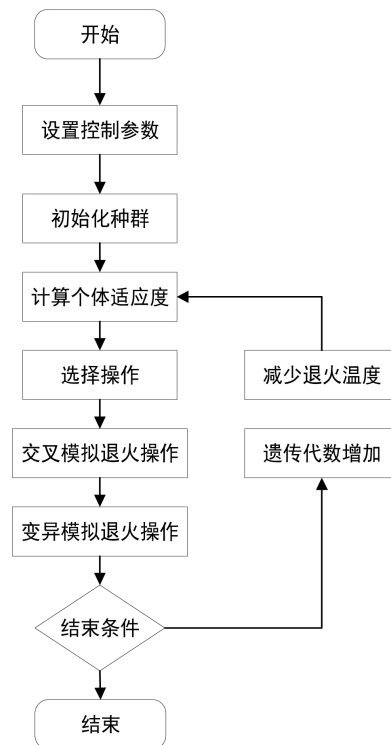


Figure 3. Algorithm flow chart
图 3. 算法流程图

步骤 3: 进行选择操作, 计算个体适应度值、评价个体好坏, 从中选择出较好的个体组成下一代种群。

步骤 4: 进行交叉操作, 得到交叉个体进行模拟退火操作, 根据 P_c 概率判断是否接受交叉操作所得到的新个体, 然后再根据 *Metropolis* 准则判断是否接受, 如果不能被接受, 重复进行交叉操作直到出现可以被接受的个体之后进入下一步。

步骤 5: 进行变异操作, 得到的变异个体进行模拟退火操作, 根据 P_m 概率判断是否接受交叉操作所得到的新个体, 然后再根据 *Metropolis* 准则判断是否接受, 如果不能被接受, 重复进行变异操作直到出现可以被接受的个体之后进入下一步。

步骤 6: 根据公式 $l_{t+1} = a \cdot l_t$ 对算法进行降温。

步骤 7: 判断是否满足终止条件, 若满足条件输出最优个体, 终止; 若不满足条件返回到步骤 3 继续寻优直到满足条件为止。

算法流程如图 3 所示。

4. 仿真比较及分析

算法运行环境为 Intel core i3、CPU 主频 2.4 GHz、4 GB 内存、Windows 10 操作系统, 并采用 Java 语言编写。算法的主要参数: 种群数取 100, 最大循环代数取 200, 初始温度为 1000℃。

4.1. 仿真试验 1

第一种算例是分别以 FT06、FT10、FT20 这三个调度问题作为测试基准[11], 将其他文献提供的算法进行实现和计算, 每类问题中均进行了 10 次仿真计算平均值, 将最优值与其他算法进行分析对比, 这些算法包括 GT-GA (Giffler-Thompson GA)、GP-GA (Generalized-Permutation GA)等。实验结果见表 1。

Table 1. Comparison table of the results of experiment 1

表 1. 实验一结果对比表

算法	FT06 (6*6)	FT10 (10*10)	FT20 (20*5)
最优解	55	930	1165
GT-GA	55	930	1184
GP-GA	55	936	1181
SB-GA	55	938	1175
P-GA	55	960	1294
本文算法	55	960	1175

对于标准算例 FT06 问题, 本算法能够快速收敛得到最优解, 对于 FT10 与 FT20 问题也能得到次优解。我们从实验结果表一中得到本文中的混合遗传算法最优解相比较好或持平与其他算法。然而, 基准问题只适用于工件工序确定的情况, 即每台机器固化了的加工工序及加工时间, 不涉及机器分配及负载问题。图 4 显示了 FT06 最优解 Gantt 图。

4.2. 仿真试验 2

第二种算例的设计使用文献[12]中提出了一个模拟车间作业的算例, 该算例中有 6 个工件在 10 台机器上加工, 并且每个工件上有 6 道加工工序的实例, 其加工时间如表 2 所示。

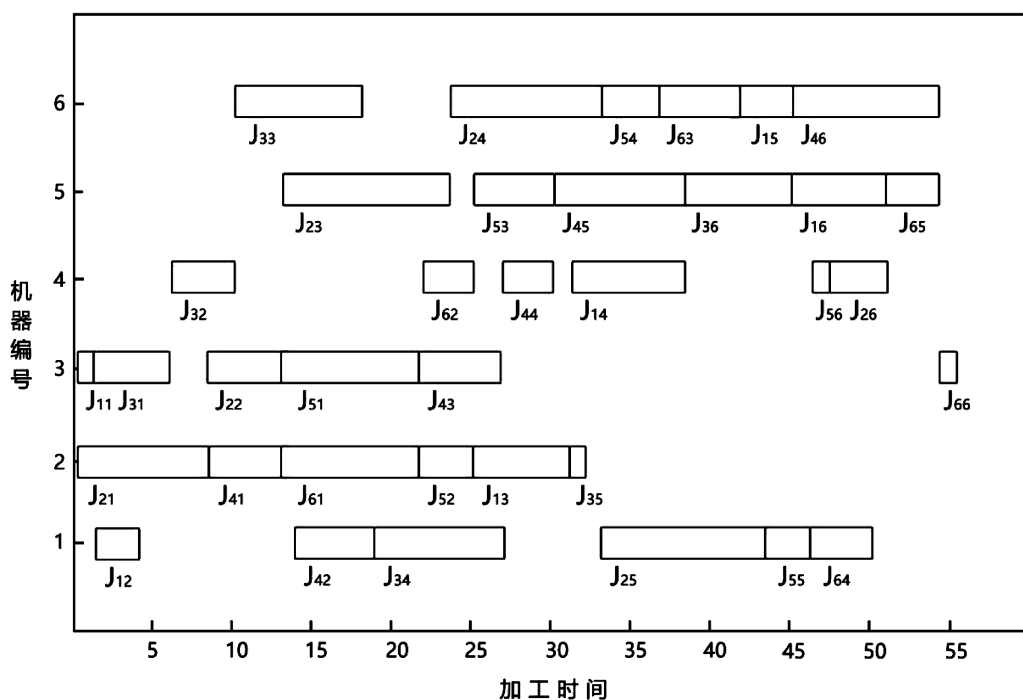


Figure 4. Optimal scheduling diagram of FT06 example
图 4. FT06 算例最优调度图

Table 2. Job shop scheduling example table
表 2. 车间调度算列表

	O	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}
J_1	O_{11}	-	-	4	-	10	-	-	12	-	15
	O_{12}	-	6	-	8	-	-	-	-	-	11
	O_{13}	-	-	1	-	-	9	-	-	4	-
	O_{14}	-	-	-	7	-	-	-	-	-	-
	O_{15}	-	-	-	-	6	-	-	-	-	-
	O_{16}	-	2	-	-	-	-	13	-	-	14
J_2	O_{21}	10	-	-	-	-	15	-	-	-	-
	O_{22}	-	-	8	-	-	-	-	6	-	-
	O_{23}	-	-	-	5	-	-	7	-	-	-
	O_{24}	4	-	-	-	-	8	-	-	3	-
	O_{25}	-	10	-	-	-	-	12	-	-	-
	O_{26}	-	-	-	4	-	-	7	-	-	-
J_3	O_{31}	-	-	9	-	-	-	-	-	-	14
	O_{32}	-	-	-	-	8	-	-	-	-	-
	O_{33}	-	9	-	-	-	5	-	6	-	-
	O_{34}	-	-	4	-	-	6	-	-	-	-
	O_{35}	-	-	7	-	-	-	-	-	-	9
	O_{36}	-	-	-	-	-	6	-	-	9	-
J_4	O_{41}	-	-	-	5	-	-	-	4	-	-
	O_{42}	-	-	6	-	-	-	5	-	-	-

Continued

	O_{43}	5	-	-	8	-	-	11	-	-	9
	O_{44}	-	3	-	-	11	12	-	5	-	10
	O_{45}	-	-	-	-	-	8	-	-	8	-
	O_{46}	1	-	-	5	-	-	7	-	-	8
	O_{51}	6	-	-	-	-	3	-	-	-	-
	O_{52}	3	-	-	7	-	6	-	-	-	8
J_5	O_{53}	-	-	9	-	-	-	-	-	-	11
	O_{54}	-	7	-	-	1	-	-	-	-	9
	O_{55}	5	-	-	14	-	-	-	-	-	-
	O_{56}	-	-	-	-	5	-	-	8	-	-
	O_{61}	-	-	-	-	10	8	-	-	-	-
	O_{62}	-	-	-	3	-	-	-	-	-	3
J_6	O_{63}	-	-	-	-	10	-	-	-	-	-
	O_{64}	-	-	-	-	-	9	-	-	-	10
	O_{65}	-	-	-	4	-	-	-	7	-	-
	O_{66}	-	-	13	-	-	15	-	-	-	14

当算法程序运行时，在原有的决策基础上再对每台设备运行的负载情况进行考虑，有利于机器分配均匀，减少机器等待时间。由表 3 得出，使用相同算例，本文算法能够求解出更好的最优值，同时反映到机器上的负载平方差更小，更利于机器分配均衡。图 5 显示了本文算法求解的车间调度问题的最优 Gantt 图。

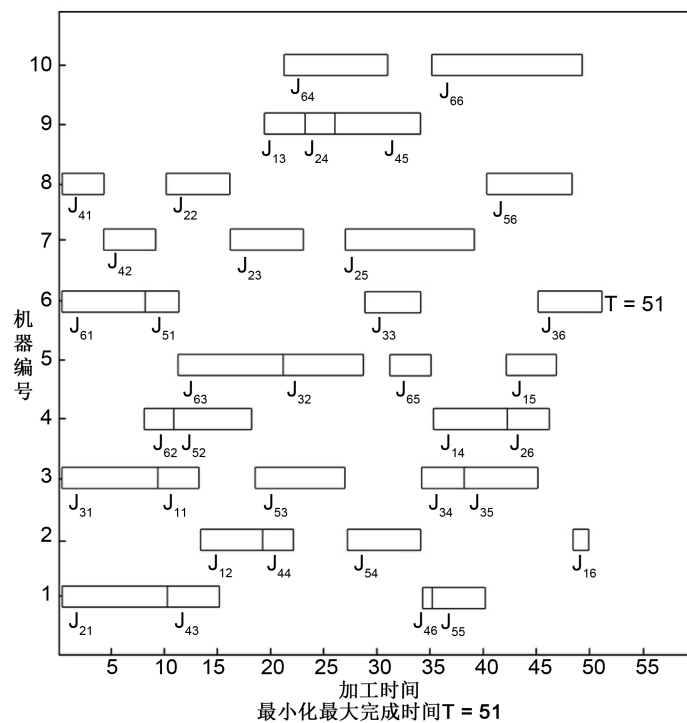


Figure 5. The optimal scheduling diagram of job shop scheduling example in experiment 2

图 5. 实验二车间调度算例最优调度图

Table 3. Test and comparison results of Job Shop scheduling example table
表 3. 车间调度算例表的测试对比结果

	C_{\max}	T_{total}	\bar{T}	N	C_{\max}
文献[12]	min	53	211	21.1	9.731
	avg	54	218	21.8	11.028
本文	min	51	223	22.3	4.954
	avg	52	226	22.6	5.687

5. 结论

平衡负荷的生产柔性作业车间调度问题是对传统柔性作业车间调度问题的改良，它更接近于实际生产调度问题，针对该问题的特点，本文提出了一种基于混合遗传模拟退火算法的平衡负荷算法。即考虑一道工序由多台设备加工的可能性，结合 SA 与 GA 算法的同时提出了一种根据机器负载因子分配加工机器的方法。通过对基准算例和实际车间应用的算例进行模拟和仿真，对算法性能进行比较分析和评价，结果表明了算法的有效性和可行性。

在实际生产中，作业车间调度通常属于具有多种约束条件的多目标优化问题，为了将本文提出的算法应用于实际生产，还需进一步深入研究。

参考文献

- [1] Bruker, P. and Schlie, R. (1990) Job Shop Scheduling with Multi-Purpose Machines. *Computing*, **45**, 369-375. <https://doi.org/10.1007/BF02238804>
- [2] 潘全科, 王文宏, 朱剑英, 等. 基于粒子群优化和变邻域搜索的混合调度算法[J]. 计算机集成制造系统, 2007, 13(2): 323-328.
- [3] 何利, 刘永贤, 谢华龙, 刘笑天. 基于粒子群算法的车间调度与优化[J]. 东北大学学报(自然科学版), 2008, 29(4): 555-568.
- [4] 黄歆雨. 基于混合遗传算法的柔性作业车间动态调度问题研究[D]: [硕士学位论文]. 福建: 福州大学, 2016.
- [5] 周鑫, 马跃, 胡毅. 求解车间作业调度问题的混合遗传模拟退火算法[J]. 小型微型计算机系统, 2015, 27(6): 184-193.
- [6] 梁旭, 黄明, 常征. 求解车间调度问题的一种新遗传退火混合策略[J]. 计算机集成制造系统, 2005, 2(36): 370-374.
- [7] 刘巍巍, 马雪丽, 刘晓冰. 面向柔性作业车间调度问题的改进邻域搜索算法[J]. 计算机应用与软件, 2015, 32(4): 234-238.
- [8] 黄学文, 张晓彤, 孙榕, 李冠雄. 基于遗传算法的工艺路径柔性调度算法[J]. 运筹与管理, 2018, 27(6): 184-193.
- [9] 赵卫. 模拟退火遗传算法在车间作业调度中的应用[J]. 计算机仿真, 2011, 28(7): 361-364.
- [10] 陈成, 邢立宁. 求解柔性作业车间调度问题的遗传——蚁群算法[J]. 计算机集成制造系统, 2011, 17(3): 615-621.
- [11] Kacem, I., Hammadi, S. and Borne, P. (2002) Approach by Localization and Multi-Objective Evolutionary Optimization for Flexible Job-Shop Scheduling Problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, **32**, 1-13.
- [12] 薛宏全, 魏生民, 张鹏, 杨琳. 基于多种群蚁群算法的柔性作业车间调度研究[J]. 计算机工程与应用, 2013, 49(24): 243-248, 261.

知网检索的两种方式:

1. 打开知网首页: <http://cnki.net/>, 点击页面中“外文资源总库 CNKI SCHOLAR”, 跳转至: <http://scholar.cnki.net/new>, 搜索框内直接输入文章标题, 即可查询;
或点击“高级检索”, 下拉列表框选择: [ISSN], 输入期刊 ISSN: 2161-8801, 即可查询。
2. 通过知网首页 <http://cnki.net/>顶部“旧版入口”进入知网旧版: <http://www.cnki.net/old/>, 左侧选择“国际文献总库”进入, 搜索框直接输入文章标题, 即可查询。

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

期刊邮箱: csa@hanspub.org