

文章编号: 1007-5429(2009)02-0059-08

改进遗传算法解决柔性作业车间调度问题

刘琼, 张超勇, 饶运清, 邵新宇

(华中科技大学机械学院数字制造装备与技术国家重点实验室, 武汉 430074)

摘要: 柔性作业车间调度问题是经典作业车间调度问题的扩展, 它允许工序在多台机器中的任意一台上加工。针对柔性作业车间调度问题的特点, 提出一种扩展的基于工序的编码及其主动调度的解码机制, 并设计一种初始解产生机制和两种有效的交叉和变异操作。为了克服传统遗传算法早熟和收敛慢的缺点, 设计了精英解保留策略和子代产生模式结合的改进遗传算法应用于该调度问题。最后运用提出的算法求解基准测试问题验证算法的有效性。

关键词: 柔性作业车间调度; 遗传算法; 交叉操作; 变异操作

中图分类号: TP301

文献标识码: A

Flexible Job-Shop Scheduling Problem with Improved Genetic Algorithm

LIU Qiong, ZHANG Chao-yong, RAO Yun-qing, SHAO Xin-yu

(State Key Laboratory of Digital Manufacturing Equipment & Technology, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: The Flexible Job-shop Scheduling problem (FJSP) is a generalization of the classical JSP, where operations are allowed to be processed on any among a set of available machines. According to the characteristics of the FJSP, an extended operation-based coding and its decoding mechanism of active scheduling is presented. Also, a mechanism of generating the initial solutions and two effective crossover operators and mutation operators are designed for the genetic algorithm. In order to preserve the good characteristics of the previous generation and reduce the disruptive effects of genetic operators, an improved genetic algorithm which combines elitist model and generation alteration model of genetic algorithm is proposed to solve the FJSP. The proposed algorithm is tested on a set of standard instance taken from the literature and compared with other approaches. The results have shown the effectiveness of the algorithms.

Key words: flexible job-shop scheduling problem; genetic algorithm; crossover operator; mutation operator

1 引言

生产调度优化是先进制造技术和现代管理技术的核心技术, 国内外许多学者进行了研究, 但是大多数研究针对的是经典作业调度问题 (Job-Shop

Scheduling Problem, JSP) 的优化。在经典 JSP 中, 每个工件的工序顺序是预先确定, 并且每道工序在指定的机器上加工。而在实际生产中, 允许工序在多台机器中的任意一台上加工, 这类问题就是柔性作业车间调度问题 (Flexible Job-Shop Scheduling

收稿日期: 2008-04-20; 修回日期: 2008-11-30

基金项目: 国家高技术研究发展计划(863)(2007AA04Z107, 2007AA04Z190); 国家自然科学基金(70772056)

作者简介: 刘琼(1965-), 女, 重庆人, 博士, 副教授, 主要研究方向为物流管理、网络协同设计与制造。

Problem, FJSP)。FJSP 减少了机器约束,扩大了可行解的搜索范围,增加了问题的难度。

FJSP 是经典 JSP 的扩展,它不仅需要确定工序加工的顺序,还要给每个工序分配机器,因此是比 JSP 更复杂的 NP-hard 问题^[1]。目前,解决的方法主要可归为两类:精确方法和近似方法。精确方法包括分支定界法、整数规划法等,但只能求解较小规模的 FJSP 问题。近似方法包括优先分配规则法^[2]、遗传算法^[3-5]、禁忌搜索算法^[6,7]和模拟退火算法^[8]等,能较快地得到问题的较优解,满足实际问题的需要。目前求解 FJSP 的近似方法可以分为两类:分级方法和集成方法。分级方法的原理主要基于每道工序的加工机器确定后,FJSP 就转化为经典的 JSP^[7];集成方法是将分配工序机器(路径柔性)和确定工序在机器上的加工顺序同时考虑,该方法难度较大,但是一般能得到更好的结果^[6]。

遗传算法借鉴了“物竞天择、适者生存”的进化准则,具有隐含并行性和全局解空间搜索的特点,在生产调度领域得到广泛的应用。但是 FJSP 问题不仅要给每个工序分配加工机器,还需要确定每台机器上工序的加工顺序,比传统的调度问题更具复杂性。尽管目前国内外已有许多学者进行了研究,但如何利用遗传算法高效求解 FSJP,仍被认为是一个具有挑战意义的课题。本文针对 FJSP 的特点,提出了有效的编码、解码和初始解产生机制,设计了两种交叉和变异操作,运用一种改进遗传算法解决不同目标的 FJSP 问题,并通过求解基准实例验证算法的有效性。

2 柔性作业车间调度问题的描述

柔性作业车间调度问题可描述如下: 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。此外,在加工过程中还需满足以下约束条件。

- (1) 每台机床一次只能加工一个工件;
- (2) 工序一旦进行不能中断;
- (3) 假定工件之间具备相同的优先级;
- (4) 不同工件的工序之间没有先后约束。

一个 3 个工件在 5 台机器上加工的柔性作业车间调度实例如表 1 所示。本文考虑以下常用的性

能指标:最大完工时间 C_{\max} (Makespan) 最小、机器最大负荷 (Workloads) 最小和总拖期 (total tardiness) 时间最小,在实际应用中可针对具体问题要求选择相应的性能指标。

表 1 柔性作业车间调度问题的加工时间表

工件	工序	加工时间				
		M_1	M_2	M_3	M_4	M_5
J ₁	O_{11}	3	—	5	4	6
	O_{12}	—	3	2	9	—
	O_{13}	2	—	4	2	3
J ₂	O_{21}	2	3	4	3	—
	O_{22}	—	5	6	4	4
	O_{23}	—	5	3	—	3
J ₃	O_{31}	—	—	3	6	—
	O_{32}	—	2	5	—	4
	O_{33}	—	4	5	2	4

对于 n 个工件、 m 台机器的柔性作业车间调度问题,这三种性能指标如下。

(1) 设 C_i 是工件 J_i 的完工时间,则最大完工时间 C_{\max} 最小的目标函数为: $\min \{ \max C_i, i = 1, \dots, n \}$;

(2) 设 W_j 是机器 M_j 上的负荷(或机器 M_j 上的工作量),则机器上最大负荷 W_M 最小的目标函数为: $\min \{ \max W_j, j = 1, \dots, m \}$;

(3) 设 D_i 为工件 J_i 的交货期,则工件总拖期时间最小的目标函数为: $\min \sum_{i=1}^n (\max \{ C_i - D_i, 0 \})$ 。

3 基于改进遗传算法的柔性作业车间调度问题

3.1 初始解

由于 FJSP 问题的复杂特性,初始解质量对于算法求解 FJSP 有很大影响,随机的初始解群体通常提供较差质量的解。目前初始解一般采用 Kacem 等提出的局部(localization)化方法^[4],这种方法同时考虑每台机器的最小加工时间或总加工时间产生调度解,但是该方法产生的解缺乏多样性,可能使算法早熟而无法求出最优解。本文提出一种简单实用的初始解产生策略:首先随机产生工序的加工顺序,然后在每个工序的机器集中随机选择两台机器,如果产生的随机值(在 0~1 之间随机产生)小于 0.8,则选择加工时间短的机器,否则选择加工时间长的机器。测试结果显示这种初始解产生机制不仅能达到更好效果,并且增加了解的多样性。

3.2 编码和解码

编码是遗传算法成功实施优化的首要和关键问题。对于传统的 JSP 现在应用较多的是基于工序的编码,它具有任意置换染色体后总能得到可行调度、避免死锁、对解空间表征的完全性等和解码成主动调度等优点。基于工序编码染色体中每个工件的工序都用相应的工件序号表示,从左到右扫描染色体,第 k 次出现的工件序号,表示该工件的第 k 道工序。

但是 FJSP 问题不仅要确定工序的加工顺序,还需为每道工序选择一台合适的机器,仅采用基于工序的编码方法不能得到问题的解。本文设计了一种扩展的基于工序的编码,该编码由两部分组成,第一部分为基于工序的编码,用来确定工序的加工先后顺序;第二部分为基于机器的编码,用来选择每道工序的加工机器。须注意基于机器的编码中机器顺序是按照每个工件的工序加工次序排列,没有与基于工序的编码对应,这种基于机器的编码一方面能确保交叉后产生可行解,另一方面也有利于工序确定其加工机器。融合这两种编码方法,可得到柔性作业车间调度问题的一个可行解。

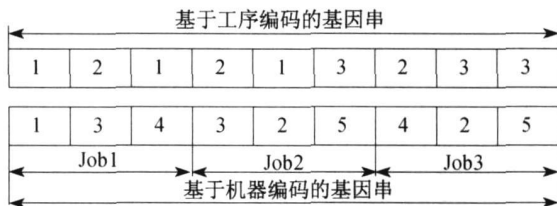


图1 扩展基于工序的编码

实例表 1 的一个扩展的基于工序编码的染色体如图 1 所示,该染色体由基于工序编码的基因串和基于机器编码的基因串组成。其中基于工序编码的基因串确定工序的加工先后顺序(或每台机器上的排序),基于机器编码的基因串确定每个工件工序的加工机器。在图 1 中的染色体表示的工序及其加工机器的顺序序列为 $(O_{11}, M_1), (O_{21}, M_3), (O_{12}, M_3), (O_{22}, M_2), (O_{13}, M_4), (O_{31}, M_4), (O_{23}, M_5), (O_{32}, M_2), (O_{33}, M_5)$, 加工时间序列为 $[3\ 4\ 2\ 5\ 2\ 6\ 3\ 2\ 4]$ 。

解码是将染色体转化为调度解的过程。设 u 为染色体任意一个工件的工序 O_{ij} , 加工时间为 p_u , 且工序 u 的开始加工时间为 ST_u , 则完工时间为 $ST_u + p_u$, 设工序的工件紧前工序和机器紧前工序分别为 $JP[u]$ 和 $MP[u]$ (如果它们存在), 那么工序 u 的开始加工时间由它的工件紧前工序和机器紧前工序中完工时间的最大值决定。所有工件从 0 时刻开始加工,

在解码过程中工序 u 开始加工时间的计算公式为

$$ST_u = \max(ST_{JP[u]} + p_{JP[u]}, ST_{MP[u]} + p_{MP[u]}) \quad (1)$$

在动态环境下工件 J_i 的释放时间 R_i 不等于零, 则 J_i 的第一道工序的开始加工时间由工序的释放时间和该工序机器前任工序的完工时间中的最大值决定, 非第一道工序的开始加工时间仍可根据公式 (1) 计算。工件 J_i 的第一道工序的开始时间 ST_u 计算公式为

$$ST_u = \max(R_i, ST_{MP[u]} + p_{MP[u]}) \quad (2)$$

可以看出, 转化染色体为工序及其加工机器序列后, 可依据式 (1) 和式 (2) 进行解码, 得到该染色体对应的半主动调度解。此外, 本文将一种插入式贪婪解码算法引入到 FJSP, 使染色体经过解码后产生主动调度。该插入式贪婪解码算法解码方法如下。

首先将染色体看作工序的有序序列, 根据工序在该序列上的顺序进行解码, 将每道工序插入到该工序可选加工机器上的最佳可行的加工时刻安排加工, 按此方式直到序列上所有工序都安排在其最佳可行的地方。应用该插入式贪婪解码算法解码图 1 中染色体得到图 2 所示的机器甘特图。

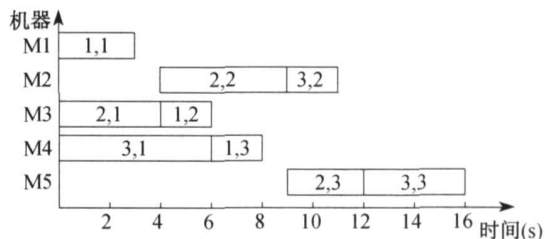


图2 染色体解码得到调度解的甘特图

3.3 交叉操作

交叉操作是将种群中两个个体随机地交换某些基因, 期望将有益的基因组合在一起。针对 FJSP 不仅需要确定工序的加工顺序, 还要给工序分配机器的特点, 本文提出了两种交叉操作, 第一种是改进 POX 交叉操作设计的 IPOX 操作, 用于染色体中工序的加工顺序的交叉^[9]; 第二种是多点交叉操作 MPX (Multi-point Preservative Crossover), 用于染色体工序分配的机器的交叉。下面对这两种交叉分别进行介绍。

(1) IPOX 交叉操作

IPOX 交叉操作仅交叉父代染色体中工序的加工顺序, 保留工件中工序分配的机器到子代。交叉操作过程如图所示, 其中 P_1 和 P_2 为表 1 中调度问题的两个父代染色体, 交叉产生子代 C_1 和 C_2 。IPOX 交叉操作的过程为: 把所有的工件机随机分

成两个集合 J_1 和 J_2 ,复制 P_1 包含在 J_1 的工件到 C_1 , P_2 包含在 J_2 的工件到 C_2 ,保留它们的位置,复制 P_2 包含在 J_2 的工件到 C_1 , P_1 包含在 J_1 的工件到 C_2 ,保留它们的顺序。

C_1	1	2	3	1	1	2	2	3	3
	↑		↑	↑	↑			↑	↑
P_1	1	2	3	1	1	2	2	3	3
P_2	2	1	3	1	2	3	2	1	3
	↓				↓		↓		
C_2	2	1	3	1	2	1	2	3	3

$J_1 = \{1, 3\}, J_2 = \{2\}$

图 3 基于工序编码的 IPOX 交叉操作

(2) MPX 交叉操作

MPX 操作仅交叉父代染色体中工序选定的机器,工序的加工顺序保留到子代,MPX 交叉操作过程如图 4 所示。对于表 1 调度问题,设父代 P_1 和 P_2 交叉产生子代 C_1 和 C_2 。MPX 交叉操作的过程为:首先随机产生一个由整数 0、1 组成与染色体长度相等的集 $Rand^0_1$,依次在 P_2 中选出 P_1 与集 $Rand^0_1$ 中的 1 位置对应相同的工序,交换它们分配的机器, P_1 和 P_2 中其他机器的保留到子代,这样分别产生子代 C_1 和 C_2 。因为是同一工序加工机器的交换,产生的子代染色体必为可行解。

	Job1	Job2	Job3
C_1	1 2 1	2 4 2	3 3 4
	↑	↑ ↑	↑
P_1	1 4 1	2 3 5	3 2 4
$Rand^0_1$	0 1 0	0 1 1	0 1 0
P_2	4 2 3	3 4 2	4 3 3
	↓	↓ ↓	↓
C_2	4 4 3	3 3 5	4 2 3

图 4 基于机器编码的 MPX 交叉操作

3.4 变异操作

变异操作的目的是维持群体多样性,防止出现早熟现象。本文设计了两种变异操作,一种是扩展的插入变异操作,即在基于工序编码的染色体中随机选择一个工件的工序(基因),将它插入到一个随机的位置,保持所有工件工序分配机器的不变,扩展的插入变异如图 5 所示。假设 P_1 染色体位置 5 的工件 2 插入到位置 3 的工件 3 之前,得到可行解的子代 C_1 ,保持染色体工序分配的机器不变。第二种分配机器的变异操作,即随机选择基于机器编码染色体的两个基因,在工序的加工机器集中随机选取机器替换该基因,该变异操作如图 6 所示。

P_1	2	1	3	1	2	3	2	1	3
C_1	2	1	2	3	1	3	2	1	3

图 5 基于工序编码的插入变异操作

	Job1	Job2	Job3
P_1	1 4 1	2 3 5	3 2 4
C_1	1 4 4	2 3 2	3 2 4

图 6 基于机器编码的变异操作

3.5 选择操作

在遗传算法中,选择是根据对个体适应度的评价从种群中选择优胜的个体,淘汰劣质的个体。本文改进遗传算法选择操作采用锦标选择,锦标选择就是从种群中随机选择两个个体,如果随机值(在 0~1 之间随机产生)小于给定概率值 r (概率值 r 是一个参数,一般设置为 0.8),则选择优的一个,否则就选择另一个。被选择的个体放回到种群,可以重新作为一个父染色体参与选择。

此外,针对柔性作业车间调度问题的特点,本文设计了一种精英解保留策略,该策略是将父代群体中最优的 1% 个体不参加遗传操作直接复制到下一代中。

3.6 改进遗传算法求解柔性作业车间调度问题

在传统的遗传算法中,交叉产生的子代总是被接受,即使它们的适应度远低于父代的适应度,这可能造成优良解丢失或破坏,导致传统遗传算法求解调度问题并不很成功,它易于早熟和收敛于较差解。在柔性作业车间调度问题中,染色体分别由基于工序编码和基于机器编码的基因串两部分组成,交叉操作对算法的影响更大。根据 FJSP 问题既要同时确定工序的加工顺序及为每道工序选择机器的特点,本文设计了一种子代产生模式的改进遗传算法(Improved Genetic Algorithm,IGA)用于 FJSP 问题。在改进遗传算法中,针对父代中一种编码的基因串,两父代交叉 n 次生成 $2n$ 后代,在父代和所有 $2n$ 个后代中选择最优的两个染色体作为下一代,这样既能将最优染色体保留到下一代,又能保证子代的多样性;这与人类繁殖类似,使父代的优良特性更好地传递到下一代。针对 TSP 和 JSP 问题的测试显示,与传统的遗传算法比较,应用子代产生模式的遗传算法在收敛速度和解的质量方面都有较大提高^[10]。改进遗传算法具体步骤如下。

Step¹ 始化随机产生 P 个染色体个体, P 为种群规模。

Step² 算个体适应度, 评价个体适应度值。

Step³ 断是否达到终止条件, 若满足则输出最好解, 结束算法; 否则转 Step⁴。

Step⁴ 选择策略选取下一代种群。

Step^{5a} 按交叉概率 P_c , 对两父代根据选择的交叉操作交叉 n 次, 从父代和所有后代中选择最优两染色体作为下一代。

Step^{5b} 按变异概率 P_m 选择个体, 根据选择的变异操作生成新个体。

Step⁶ 成新一代的种群, 返回到 Step³。

4 计算结果与分析

上述改进遗传算法用 C++ 编程, 程序运行环境为 P4 CPU, 主频 3.0G, 内存为 512MB。为了评估和比较提出算法的性能, 首先选取了研究较普遍的 Kacem 和 Brandimarte 基准问题, Kacem 基准问题包括两个 8×8 和 10×10 的实例^[11]; Brandimarte 基准问题包括 10 个工件数从 10 到 20 之间, 机器数从 4 到 15 之间的实例, 实例数据可在网站 <http://www.dsia.ch/monaldo> 下载。

然后, 通过对模拟的实际问题进行测试, 验证算法的有效性。

改进遗传算法的运行参数设置如下: 种群规模 $P=300$, IPOX 交叉概率 45%, MPX 交叉概率 45%, 交叉次数等于 $n+m$, 两种变异操作的变异概率各为 2%, 选择概率值 $r=0.8$, 最大进化代数 100。

4.1 基准问题测试

对于 Kacem 和 Brandimarte 基准问题, 为了得到算法求解的最好解、平均解和平均计算时间, 使用改进遗传算法连续求解每个实例 10 次。在表 2 和表 3 中分别显示了 Kacem 和 Brandimarte 实例测试结果的比较, 其中对于 Kacem 实例分别给出了与传统 GA、Kacem 等、Zhang 等遗传算法在不同性能指标函数情况下测试结果的比较^[4,5]; 对于 Brandimarte 基准实例分别给出了与 Chen、Mastrolilli 和 Gambardella 测试结果的比较^[3,6], 这些指标函数包括最大完工时间 (C_{\max}) 最小、每台机器上最大负载 W_M 最小。此外, 图 7 显示了得到实例 Mk04 最大完工时间 Makespan 等于 60 的机器甘特图。

表 2 对于 Kacem 实例的测试结果的比较

实例	目标函数	传统 GA	Kacem	Zhang 等 moGA	本文 IGA		
					$Best(C_{\max})$	$Av(C)$	$Av(T)$
8×8	C_{\max}	16	16	15	14	17.2	16.2
	W_M	14	14	14	11	13.6	10.3
10×10	C_{\max}	7	7	7	7	8.1	19.9
	W_M	7	6	5	5	6.9	12.0

表 3 对于 Brandimarte 基准实例测试结果的比较

实例	$n \times m$	(LB, UB)	Chen 的 GA	M.G. 的 TS		本文		IGA	$Av(T)$
			$Best(C_{\max})$	$Best(C_{\max})$	$Av(C)$	$Best(C_{\max})$	$Av(C)$	W_M	
Mk01	10×6	(36, 42)	40	40	40.0	40	40	36	2.2
Mk02	10×6	(24, 32)	29	26	26.0	26	27.2	26	7.4
Mk03	15×8	(204, 211)	204	204	204.0	204	204	204	0
Mk04	15×8	(48, 81)	63	60	60.0	60	63.9	60	14.2
Mk05	15×4	(168, 186)	181	173	173.0	173	173	172	14.1
Mk06	10×15	(33, 86)	60	58	58.4	62	65.8	48	32.2
Mk07	20×5	(133, 157)	148	144	147.0	140	142.2	139	28.5
Mk08	20×10	(523, 523)	523	523	523.0	523	523	523	0.6
Mk09	20×10	(299, 369)	311	307	307.0	307	312.2	299	61.2
Mk10	20×15	(165, 296)	212	198	199.2	215	226.1	189	90.4

从表 2 中可以看出, 对于 Kacem 的 8×8 和 10×10 测试实例, 改进遗传算法(IGA)在两个性能指标上得到的最优解优于或等于其他三个遗传算法的结果, 例如: 对于 8×8 实例时, 得到 C_{\max} 指标最优

值 14 和和 W_M 指标最优值 11 均优于其他三个算法获得的最优值。从表 3 可以看出, 本文 IGA 得到最优解有 5 个优于 Chen 等提出遗传算法, 只有 2 个比 Chen 等的算法差, 其他相同。从改进遗传算法

(IGA)与其他遗传算法比较,显示了 IGA 求解柔性作业车间调度问题的有效性。Mastrolilli 和 Gambardella 提出禁忌搜索算法是目前解决以最大完工时间最小为性能指标 FJSP 最先进的智能优化

算法,本文 IGA 算法得到实例 Mk07 的最优解优于 M·G·提出的算法获得的结果,并且遗传算法解决 FJSP 具有较高的柔性,可用于解决其它性能指标的问题,如交货期等指标。

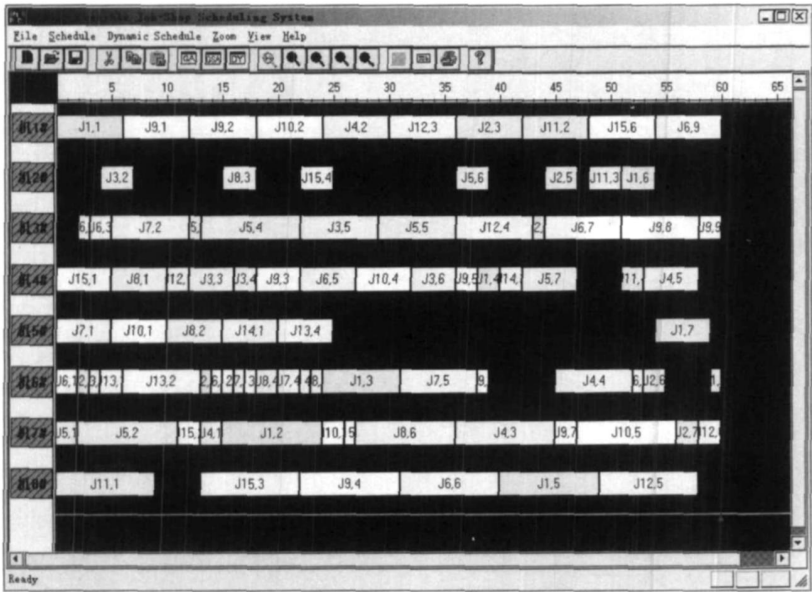


图 7 Mk04 实例的 C_{\max} 指标最小 ($Makespan=60$) 解的机器甘特图

4.2 实例仿真

通过模拟某模具厂加工车间的实际数据并取整处理,得到 12 个工作在 10 台机器上加工的 FJSP 实例数据如表 4 所示。从表 4 中可以看出,这批工件的工件到达时间不相同,并且有交货期要求,例如

工件 J1、J2、J5、J6、J7 和 J8 要求在时刻 60 前交货,其它工件要求在时刻 80 前交货,因此需要考虑总拖期最小为指标的柔性作业车间调度。使用本文改进的遗传算法求解该实例,得到总拖期为 6 时间单位的最优解机器甘特图如图 8 所示。

表 4 具有工件到达时间和交货期的 FJSP 实例

工件	释放时刻/h	工序	加工时间/h										交货期/h
			M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	
J1	7	O ₁₁	—	10	—	12	—	18	9	—	19	—	60
		O ₁₂	15	10	—	—	17	—	—	16	17	—	
		O ₁₃	—	—	—	11	—	22	10	—	—	24	
J2	8	O ₂₁	17	—	16	10	21	—	—	10	—	—	60
		O ₂₂	—	11	—	—	19	14	—	—	8	12	
		O ₂₃	—	21	—	15	—	—	25	—	—	—	
J3	10	O ₃₁	—	—	—	—	18	—	9	—	—	—	80
		O ₃₂	—	—	—	14	9	11	10	—	12	15	
		O ₃₃	—	8	—	7	10	—	—	10	—	9	
		O ₃₄	—	12	—	—	—	—	17	—	—	14	
		O ₃₅	—	18	23	17	—	12	—	16	—	23	
		O ₃₆	21	—	20	9	22	12	—	—	—	—	
J4	15	O ₄₁	—	11	10	—	8	—	9	—	7	—	80
		O ₄₂	—	17	—	—	—	15	18	—	—	18	
		O ₄₃	—	—	—	10	—	12	12	10	—	—	
		O ₄₄	20	—	11	8	—	—	9	—	—	8	
J5	19	O ₅₁	—	—	24	10	16	—	—	—	15	—	60
		O ₅₂	8	—	—	17	—	12	—	—	—	—	

续表

J6	16	O ₆₁	—	—	—	—	—	19	22	11	—	20	60
		O ₆₂	—	—	18	10	—	—	19	—	12	—	
		O ₆₃	18	—	17	16	16	12	—	—	19	—	
J7	16	O ₇₁	—	16	16	—	17	16	—	—	—	—	60
		O ₇₂	—	20	—	—	—	—	22	12	22	—	
		O ₇₃	—	18	—	—	—	12	23	14	—	—	
		O ₇₄	24	10	—	—	—	21	—	—	16	—	
J8	19	O ₈₁	21	—	—	—	—	19	21	—	11	—	60
		O ₈₂	—	—	—	—	—	—	—	10	25	11	
		O ₈₃	20	25	—	—	—	16	—	—	24	18	
J9	6	O ₉₁	—	14	9	—	—	—	—	14	—	12	80
		O ₉₂	—	22	11	21	—	—	25	—	24	—	
		O ₉₃	19	16	—	—	—	—	—	12	—	12	
		O ₉₄	—	—	—	8	7	—	10	—	—	6	
J10	23	O _{10,1}	—	8	—	6	11	—	10	—	9	12	80
		O _{10,2}	23	—	—	—	—	—	21	16	—	19	
		O _{10,3}	12	24	—	—	—	—	—	—	—	18	
J11	4	O _{11,1}	—	8	—	6	11	—	10	—	9	12	80
		O _{11,2}	23	—	—	—	—	—	21	16	—	19	
J12	21	O _{12,1}	—	8	—	6	11	—	10	—	9	12	80
		O _{12,2}	23	—	—	—	—	—	21	16	—	19	

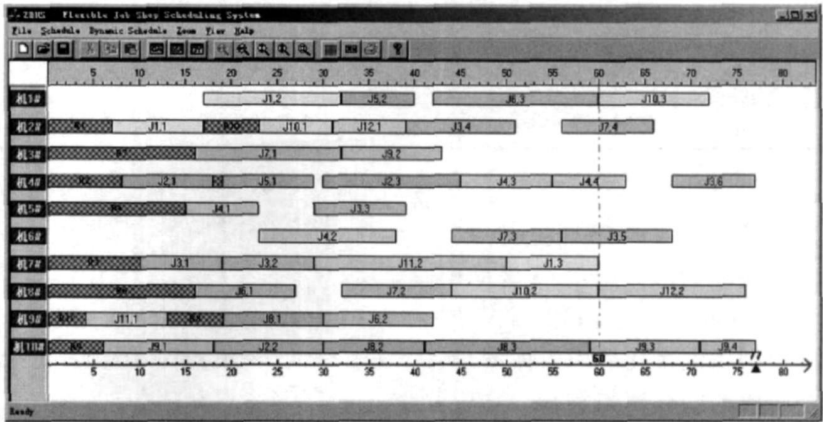


图8 总拖期为6时间单位的最优解机器甘特图

5 结语

FJSP 是比 JSP 更为复杂的组合优化问题,它不仅要确定工序的加工顺序,还要给每个工序分配机器。本文针对 FJSP 的特点,提出了一种扩展的基于工序编码,在解码过程中引入插入式贪婪解码算法,以生成主动调度;根据染色体的结构设计了两种有效的交叉和变异操作。为了克服传统遗传算法求解调度问题的不足,设计了精英解保留策略和子代产生模式结合的改进遗传算法应用于柔性作业车间调度问题。最后,使用改进的遗传算法求解基准实例和实际问题,证实了提出算法的有效性。

参考文献:

[1] Garey E L, Johnson D S, Sethi R. The complexity of flowshop and job-shop scheduling [J]. Mathematics of Operations Research, 1976, 1, 117-129.

[2] Montazeri M, Van W. Analyses of scheduling rules for an FMS [J]. Int. J. Prod. Res., 1990, 28, 785-802.

[3] Chen H, Ihlow J, Lehmann C. A Genetic Algorithm for Flexible Job-Shop Scheduling [R]. IEEE International Conference on Robotics and Automation. Detroit, 1999: 1120-1125.

[4] Kacem S, Hammadi, Borne P. Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems[R]. IEEE Transaction Systems, Man, and Cybernetics-Part C, 2002, 32(1), 1-13.

[5] Zhang H P, Gen M. Multistage-based genetic algorithm for flexible job-shop scheduling problem [J]. Journal of Complexity, 2005, 11: 223-232.

[6] Mastrolilli M, Gambardella L M. Effective neighbour-hood functions for the flexible job shop problem [J]. Journal of Scheduling, 2000, 3(1): 3-20.

[7] Barnes J W, Chambers J B. Flexible Job Shop Scheduling by Tabu Search, Graduate Program in Operations Research and Industrial Engineering[M]. Tech. Rep. ORP 9609, University of Texas, Austin, 1996.

[8] Najid N M, Dauzere-Pere S, Zaidat A. A modified simulated annealing method for Flexible Job Shop Scheduling Problem [J]. In Proceedings of the 2002 International Conference on Systems, Man and Cybernetics. Piscataway, NJ: IEEE Press, 2002: 6-12.

[9] 张超勇, 饶运清, 李培根. 基于 POX 交叉的遗传算法求解 Job-Shop 调度问题[J]. 中国机械工程, 2004, 15(23): 2149-2153.

[10] Ono I, Yamamura M, Kobayashi S. A Genetic Algorithm for Job-shop Scheduling Problems Using Job-based Order Crossover. Proc. of ICEC '96, Nagoya, 1996.

[11] Kacemi I, Hammadi S, Borne P. Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems [C] // IEEE Transactions on Systems, Man and Cybernetics, Part C, 2002, 32(1): 408-419.

(上接第 58 页)

表 2 决策变量值

X_{12}	X_{23}	X_{33}	X_{42}	X_{52}
40	50	50	60	80
X_{62}	Y_{22}	Y_{32}	Z_{21}	Z_{22}
100	246	88	30	30
Z_{23}	T_{21}	L_{11}	L_{21}	L_{31}
197.142	40.128	6	0.05	39.42840
q_2^2	q_3^2	q_2^2	q_3^2	q_1^n
1	1	1	1	1
q_2^n	q_3^n	其他		
1	1	0		

5 结语

下一步的研究重点主要为两方面:(1) 考虑模型中的关键影响因素,进行再优化设计;(2) 将不同地区的电子产品污染控制条例量化后引入模型,从而更准确地确定每个阶段的废弃率与翻新率,以及在能量回收阶段所需的额外费用。

参考文献:

[1] Home Appliance Recycling Law, Policy Planning Division, Waste Management and Recycling Department[S]. Ministry of the Environment, Tokyo, Japan, 2001.

[2] Yoon H, Jang Y. The Practice and Challenges of Electronic Waste Recycling in Korea with Emphasis on Extended Producer Responsibility (EPR)[C]. Proceedings of the IEEE International Symposium on Electronics and the Environment, May 2006: 326-330.

[3] Waste Electrical and Electronic Equipment [EB/OL]. http://ec.europa.eu/environment/waste/weee/index_en.htm

[4] 电子信息产品污染控制管理办法(第 39 号)[S]. 中华人民共和国信息产业部, 2006.

[5] Producer Take-back. “EPR” Extended Producer Responsibility Creates the Incentive for Green Design [EB/OL]. http://www.etoixics.org/site/PageServer?pagename=svtc_extended_producer_responsibility.

[6] Product Return and Recycling [EB/OL]. <http://www.hp.com/hpinfo/globalcitizenship/environment/return/index.html>.

[7] Apple and Environment [EB/OL]. <http://www.apple.com/environment/>

[8] Dell and Environment [EB/OL]. http://www.dell.com/content/topics/global.aspx/corp/environment/en/recycling_main?c=us&l=en&s=corp

[9] IEEE Standard 1680, IEEE Standard for Environmental Assessment of Personal Computer Products, Including Laptop Personal Computers, Desktop Personal Computers, and Personal Computer Monitors [R], IEEE Computer Society, New York, April 2006.

[10] Fleischmann M, Quantitative Models for Reverse Logistics [M]. Springer 2001, 58-86.

[11] 马祖军, 代颖. 产品回收逆向物流网络优化设计模型[L]. 管理工程学报, 2005, 19(4).

[12] 冷杰, 熊寿刚. 再制造逆向物流网络选址模型研究[L]. 物流技术, 2005(6).

[13] 马祖军, 张殿业, 代颖. 再制造逆向物流网络优化设计模型研究[L]. 交通运输工程与信息学报, 2004, 2(2).

[14] Guo W, Jiang X. Green Manufacturing [C]. 2nd International Eco-electronics Conference, December 7—8th, 2006.

[15] U.S. Department of Commerce Technology Administration Office of Technology Policy, Recycling Technology Products—An Overview of E-Waste Policy Issues[R]. July 2006.

[16] Hicks C, Dietmar R, Eugster M. The recycling and disposal of electrical and electronic waste in China—legislative and market responses[J]. Environmental Impact Assessment Review, July 2005, 25(5): 459-471.

[17] Ecobalance. Analysis of Five Community Consumer/residential Collection of End-of-Life Electronic and Electrical Equipment[R]. April 1999.

[18] Realf M J, Raymond M, Ammons J C. E-waste: an Opportunity[N]. Materials Today, 2004: 40-45.