

# 带工艺约束并行机调度问题的一种新的遗传算法

尹文君, 刘 民, 吴 澄  
(清华大学自动化系, 北京 100084)

摘 要: 以最小化拖期任务数为目标, 研究了解决一类带工艺约束并行机调度问题的新的遗传算法. 基于向量组的染色体编码方法简单、译码快速, 并能自动满足工艺约束; 扩展的顺序交叉算子 EOX 能自动满足工艺约束并尽量保留父代遗传信息, 在较大程度上优于传统的交叉算子; 位变异和交换变异相结合的变异方法有利于更好保持种群的多样性. 大量模拟数据与生产线应用实例表明本文所提的基于向量组编码的遗传算法是相当有效的.

关键词: 调度; 并行机; 工艺约束; 遗传算法; 向量组编码

中图分类号: TP18; O221 文献标识码: A 文章编号: 0372-2112 (2001) 11-1482-04

## A New Genetic Algorithm for Parallel Machine Scheduling with Process Constraint

YIN Wenjun, LIU Min, WU Cheng

(Department of Automation, Tsinghua University, Beijing 100084, China)

Abstract: A new genetic algorithm is studied for solving parallel machine scheduling problems with process constraint to minimize the total number of tardy jobs. A so called vector group coding method is presented, which shows the quality of coding simply, decoding fast and satisfying process constraints automatically. A new crossover operator named Extended Order Crossover (EOX) is then proposed, which has the merits of automatically satisfying procedure constraints and preserving much genetic information. Mutation, the other genetic operator, is implemented with the combination of bit mutation and swap mutation to keep the population diverse. The algorithm behaves more efficiently than others experimentally using various random data and application instance from practical production line.

Key words: scheduling; parallel machine; process constraint; genetic algorithm; vector group coding

### 1 引言

带工艺约束的并行机调度 (Parallel Machine Scheduling Problems with Process Constraint, CPSP) 源于实际工业生产线模型, 每台机器可加工的工件种类受工艺约束, 而且各机器加工能力不尽相同, 需要在满足工艺约束的前提下同时确定工件加工路径以及单机上工件的加工顺序.

以往对并行机调度问题的研究一般集中在机器加工能力完全相同的调度类型上, 主要包括启发式<sup>[1]</sup>以及其它寻优方法, 其中遗传算法 (Genetic Algorithm, GA) 以其操作简单、不受问题空间特殊性限制和全局寻优能力强等优点已逐渐成为并行机调度研究的热点<sup>[2-6]</sup>, 但对 CPSP 的应用研究还不太多. 为此, 本文以最小化拖期任务数为目标, 针对一类带工艺约束并行机调度问题研究了一种新的遗传算法: 首先根据 CPSP 特点设计了一种向量组编码方法, 该方法编码简单、译码快速并自动满足工艺约束, 可使交叉、变异等遗传操作简单化; 在此基础上提出的扩展顺序交叉算子 EOX (Extended Order Crossover) 吸收了传统交叉算子的遗传优势, 而且不涉及工艺约束信息, 可操作性强; 变异采用交换变异和位变异相结合的

方法实现, 以保证种群多样性. 仿真实验和实际生产数据应用实例表明所提算法是相当有效的.

### 2 CPSP 问题描述

设  $N = \{1, 2, \dots, n\}$  为待加工工件集合,  $M = \{1, 2, \dots, m\}$  为并行机集合. 每个工件仅需经一道工序完成, 工件间无优先级限制. 每台机器一次只能加工一个工件, 所有能加工工件  $j$  的机器 (可行机) 集合记为  $M_j$ , ( $j = 1, 2, \dots, n$ ). 机器  $k$  上第  $r$  个加工的工件记为  $j(k, r)$ ,  $p_{j(k, r)}$  为工件  $j(k, r)$  在机器  $k$  上的加工时间. 若不考虑机器故障, 则工件  $j(k, r)$  的完工时间为

$$c_{j(k, r)} = \sum_{i=1}^r p_{j(k, i)} \quad (1)$$

设  $d_j$  为工件  $j$  的交货期, 记  $sig(j)$  为

$$sig(j) = \begin{cases} 1, & c_j - d_j > 0 \\ 0, & c_j - d_j \leq 0 \end{cases} \quad (2)$$

若  $sig(j) = 1$  则表明工件  $j$  拖期, 因此某特定调度方案  $S$  对应的拖期任务总数为

$$N_{tardy} = \sum_{j=1}^n sig(j) \quad (3)$$

收稿日期: 2000-06-29; 修回日期: 2001-02-21

基金项目: 国家自然科学基金项目 (No. 60004010) 和国家 863 计划“复杂制造系统智能优化”项目; 清华大学 985 规划基金

上述调度方案  $S$  可以表示为

$$S = \{S_1 S_2 \dots S_m\} \quad (4)$$

其中  $S_k = \{j(k, 1) j(k, 2) \dots j(k, n_k)\}$  为机器  $k$  上的工件加工序列,  $n_k$  为机器  $k$  加工工件总数,  $k = 1, 2, \dots, m$ .

结合式(1)和式(2), 式(3)等价于

$$N_{tardy} = \sum_{k=1}^m \sum_{r=1}^{n_k} sig(j(k, r)) \quad (5)$$

至此, 以拖期任务数最小化为目标的该类 CPSP 问题描述为: 在满足工艺约束的前提下, 寻找最优的并行机调度方案  $S_{opt}(S_1 S_2 \dots S_m)$ , 使拖期任务总数最小, 即

$$\min N_{tardy} = \min \sum_{k=1}^m \sum_{r=1}^{n_k} sig(j(k, r)) \quad (6)$$

### 3 遗传算法设计

本节从编码入手考虑构造适合 CPSP 特点的染色体基因串, 进而设计简单高效的交叉与变异算子, 最后给出整个遗传算法的流程.

#### 3.1 编码

遗传编码实现从问题解空间到染色体(字符串)空间的映射, 是 GA 设计的基础. 针对 CPSP 问题的特点, 本文提出向量组编码方法, 即染色体表示为

$$Ch = [g_1 g_2 \dots g_n] = \begin{bmatrix} j_1 & j_2 & \dots & j_n \\ k_1 & k_2 & \dots & k_n \end{bmatrix} \quad (7)$$

其中, 染色体长度等于待加工工件的总数  $n$ ; 每个基因采用二维向量  $g_i = \begin{bmatrix} j_i \\ k_i \end{bmatrix}$  表示,  $j_i$  为工件号,  $k_i$  为  $j_i$  的加工机器,  $k_i \in M_{j_i}$ .

染色体  $Ch$  的译码步骤为

- (1) 读取基因  $g_i = \begin{bmatrix} j_i \\ k_i \end{bmatrix}$ , ( $i = 1, 2, \dots, n$ );
- (2) 将工件  $j_i$  分配到机器  $k_i$  上;
- (3) 分配到同一机器上的工件根据其在染色体中的排列位置依次加工;
- (4) 根据式(2)计算  $sig(j_i)$ ;
- (5)  $i = i + 1$ , 循环  $n$  次结束.

从而, 染色体  $Ch$  所对应调度方案的拖期任务数为

$$N_{tardy} = \sum_{i=1}^n sig(j_i) \quad (8)$$

图 1 是  $8 \times 3$  规模 ( $n = 8, m = 3$ ) CPSP 问题染色体译码过程示意图.

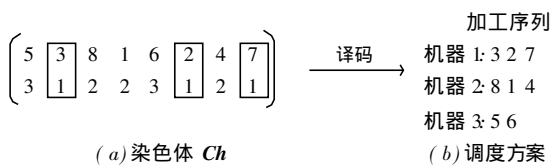


图 1 译码过程示意图

#### 3.2 交叉算子 EOX

传统交叉方法在处理约束问题时容易产生无效个体, 需额外对染色体进行可行性检验, 降低了运行速度和寻优效果.

如何设计适合约束优化(Constrained Optimization)问题的交叉算子一直是遗传算法研究的热点, 其中比较典型的是部分匹配交叉法 PMX<sup>[7]</sup> (Partially Mapped Crossover) 和顺序交叉法 OX<sup>[8]</sup> (Order Crossover), 两者的区别主要体现在对遗传基因间相互顺序的处理上: PMX 考虑基因间的绝对顺序, 而 OX 则更强调基因间的相对顺序, 两者互有优劣, 但针对 CPSP 的研究还不多. 本文基于向量组编码方法提出扩展的顺序交叉算子 EOX(Extended Order Crossover).

EOX 操作中, 两个父代个体交叉生成一个新的个体. 不妨称基因  $g_i$  中的第一行元素  $j_i$  为该基因的索引项, 则 EOX 的具体操作过程如下: 首先, 任选一父代个体  $A$ , 从其染色体中随机选择一段基因串  $s$ ; 删除另一父代个体  $B$  内与  $s$  串各基因中索引项相同的基因, 同时标出删除操作之前  $B$  中与串第一个基因索引项相同的基因所在位置  $l$ ; 最后, 将基因串  $s$  插入到个体  $B$  的位置  $l$  处, 形成新的个体. 整个交叉过程如图 2 所示.

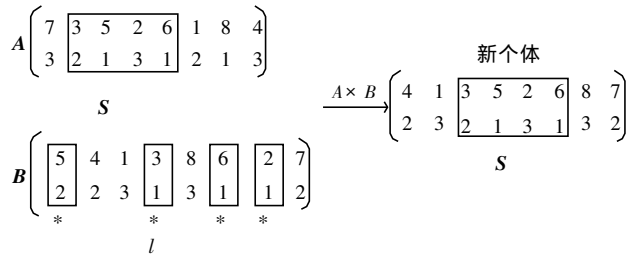


图 2 EOX 交叉示意图

由于采用向量组编码, 交叉生成的新个体自动满足工艺约束, 无需进行可行性检验; 另外, 与 PMX 和 OX 相比, EOX 同时兼顾父代染色体遗传基因间的相对和绝对顺序, 有利于遗传信息的保存, 因此性能更好.

#### 3.3 变异算子

变异采用位变异和交换变异相结合的方法:

- 位变异(bit mutation): 仅对染色体基因的第二行进行变异. 即先根据位变异概率  $p_{mut}$  选择出变异基因  $g_i$ , 再从  $j_i$  的可行机集合  $M_{j_i}$  中随机选择新的机器号代替  $g_i$  的第二行;
- 交换变异(swap mutation): 随机交换变异基因在染色体中的位置. 即先根据交换变异概率  $p_{mswap}$  选出各变异基因, 再随机交换其位置;

位变异主要改变工件的加工路径, 而交换变异则主要改变工件的加工顺序, 两者结合能较好保证种群的多样性. 显然, 这两种变异方法使新生成的个体仍然满足可行性.

#### 3.4 GA 流程

至此, 本文设计的遗传算法流程可以表示为:

Step1: 产生初始种群. 初始种群中的各个体随机生成: 首先, 以工件  $1 \sim n$  的随机排列  $\{j_1 j_2 \dots j_n\}$  作为染色体的第一行; 对工件  $j_i$ , 随机选择  $k_i \in M_{j_i}$ , 从而组成染色体的第  $i$  个基因  $g_i = \begin{bmatrix} j_i \\ k_i \end{bmatrix}$ , ( $i = 1, 2, \dots, n$ );

Step2: 选择. 采用保留最优个体的轮盘赌机制. 在最小化拖期任务数的 CPSP 问题中, 个体适应度函数采用指数定标的方法, 即

$$f(N_{tardy}) = \alpha e^{-\beta N_{tardy}} \quad (9)$$

其中,  $\alpha, \beta > 0$ ,  $N_{tardy}$  为染色体译码所得调度方案的拖期任务数.

Step3: 交叉. 采用 3.2 交叉方法;

Step4: 变异. 采用 3.3 变异方法;

Step5: 若满足停止准则, 退出; 否则, 返回 Step2. 停止准则设为最大迭代次数.

#### 4 计算实例与分析

为检验所提遗传算法的有效性, 本文进行了大量仿真实验. 首先, 比较 OX、PMX 和 EOX 等交叉算子的效果以检验采用 EOX 的遗传算法寻优性能. 进一步, 将本文所提算法、文献 [5] 基于矩阵编码的遗传算法以及常用启发式调度规则进行对比实验. 最后, 利用所提算法对某钢铁厂冷轧精整生产线的实际数据进行仿真, 结果比较满意.

##### 4.1 问题数据产生方法

根据如下规则<sup>[5]</sup>随机生成 CPSP 问题数据:

- 工件可行机集合服从均匀分布  $U[1, m]$ .
- 工件在可行机器上的加工时间满足均匀分布:  $p_i \sim U[1, ut]$ ,  $ut$  为加工时间上限;
- 各工件的交货期  $d_j$  满足均匀分布:  $d_j \sim U[1, 2 \times n \times p_{avg} / (m \times q)]$ , 其中  $n$  和  $m$  分别为工件与机器总数,  $p_{avg}$  为工件平均加工时间, 紧缩因子  $q$  表示交货约束的松紧程度;

##### 4.2 交叉算子比较

比较 EOX 与 OX、PMX 等典型交叉算子的寻优性能, 遗传算法其它结构相同, 遗传参数采用经典优化参数<sup>[9]</sup>: 种群大小: 60; 交叉率: 0.95; 位变异率: 0.01; 交换变异率: 0.4; 迭代次数: 1000;  $\alpha = 0.9$ ,  $\beta = 0.4$ . 仿真实验在 PIII550 微机上进行.

对不同规模随机 CPSP 问题进行的对比实验所得结果如表 1 所示. 其中, 每种规模均分交货期宽约束 ( $q = 0.5$ ) 和紧约束 ( $q = 1.5$ ) 两种情况, 每个问题独立运行 20 次.

表 1 采用三种不同交叉算子的 GA 运行结果比较

问题规模 ( $n \times m$ )		宽约束 ( $q = 0.5$ )				紧约束 ( $q = 1.5$ )			
		$N_{lst}$	$N_{avg}$	$N_{wst}$	S	$N_{lst}$	$N_{avg}$	$N_{wst}$	S
20 × 4	OX	0	0	0	0	0	0.1	2	0.4
	PMX	0	0	0	0	0	0.2	2	0.6
	EOX	0	0	0	0	0	0	0	0
30 × 6	OX	0	3.2	5	1.2	5 <sup>▲</sup>	7.0	9	0.9
	PMX	0	2.5	4	1.3	6	7.0	8	0.6 <sup>▲</sup>
	EOX	0	2.4	4	1.0	6	6.5	8	0.7
60 × 10	OX	2	3.8	6	1.0	10	12.3	16	1.5
	PMX	0	2.3	4	1.3	8	9.6	12	1.0
	EOX	0	2.0	3	0.9	8	9.2	11	0.9
80 × 8	OX	5	8.1	10	1.0 <sup>▲</sup>	14	16.7	18	1.2
	PMX	3	5.7	8	1.5	9 <sup>▲</sup>	13.1	15	1.4
	EOX	2	5.0	7	1.2	11	12.8	15	1.0
120 × 12	OX	2	3.4	5	0.8 <sup>▲</sup>	9	13.6	18	2.06
	PMX	0	0.5 <sup>▲</sup>	4	1.1	7	10.9 <sup>▲</sup>	14	1.68
	EOX	0	0.8	3	1.0	7	11.4	14	1.59
150 × 20	OX	4	5.5	8	1.2 <sup>▲</sup>	29	33.3	36 <sup>▲</sup>	1.7 <sup>▲</sup>
	PMX	0	3.1	7	1.8	27	34.2	38	3.0
	EOX	0	3.0	5	1.2	27	30.9	37	2.4

注:  $N_{lst}$  为 20 次计算最优值;  $N_{avg}$  为 20 次计算结果平均值;  $N_{wst}$  为 20 次计算最差值; S 为 20 次计算结果的标准差

带工艺约束的并行机调度需要同时确定工件加工路径及单机上的工件加工顺序. 采用扩展顺序交叉算子 EOX 可以尽量保存父代染色体中遗传基因间的相对和绝对顺序, 这实际上是对父代个体中工件加工路径和单机加工顺序两类信息的一部分保留, 因此比其它两种算子更适合 CPSP 问题.

表 1 用<sup>▲</sup>标出的是采用其它交叉算子得到的比 EOX 更好的结果. 可以看到, 多数情况下, 采用 EOX 算子的遗传算法运行结果都优于其它两种算法; 而且随着问题规模的增大和交货期约束的加紧, EOX 的寻优效果愈加显著, 虽然对有些规模采用 EOX 算子得到的标准差劣于其它算子, 但采用 EOX 的结果整体性能比较满意 (集中体现在平均值  $N_{avg}$  上), 因此, 基于 EOX 算子的遗传算法更适于解决 CPSP 问题.

##### 4.3 算法比较

由于 EOX 算子显示了比 PMX 和 OX 更强的寻优性能, 本文将采用基于染色体向量组编码和 EOX 交叉的遗传算法 EOX-GA 来解决最小化拖期任务数的 CPSP 问题. 其它几种比较算法包括文献 [5] 基于矩阵编码的遗传算法 (简称为 LOX-GA)、几种常用的与交货期相关的并行机调度规则 (EDD、MSLACK、CR)<sup>[9]</sup> 等.

模拟实验中, 为避免问题本身随机性造成的结果偶然性, 均针对问题规模进行对比实验. 对每种规模, 随机生成 20 组数据 (约束因子  $q = 2.0$ ); 对每组随机数据, 各算法均独立运行 20 次, 比较每种规模下的平均拖期数. 遗传算法参数设置为: 种群大小 45; 交叉率 0.95; LOX-GA 变异率为 0.01; EOX-GA 位变异率和交换变异率分别为 0.01 和 0.4, 迭代次数 600;  $\alpha = 0.9$ ,  $\beta = 0.4$ .

表 2 不同算法比较结果

算法 规模	EOX-GA	LOX-GA	EDD	MSLACK	CR
	$N_{avg}(T_{avg})$	$N_{avg}(T_{avg})$	$N_{avg}$	$N_{avg}$	$N_{avg}$
20 × 4	2.5(2.135)	3.2(4.025)	6.0	8.1	9.8
30 × 6	6.1(2.401)	6.4(4.72)	9.2	14.3	16.7
60 × 8	13.5(4.319)	16.6(7.532)	20.4	29.5	36.7
80 × 8	20.0(4.442)	24.2(9.662)	30.9	41.4	50.3
120 × 10	34.7(8.072)	38.8(15.29)	50.8	67.4	82.9
150 × 12	46.4(8.21)	51.0(14.07)	60.1	85.2	100.8

注:  $N_{avg}$  为对应规模平均拖期数;  $T_{avg}$  为每个随机问题的平均运行时间 (秒), 由于三种并行机调度规则的运行时间比较短 ( $< 0.01$  秒), 表中仅列出两类遗传算法的平均运行时间.

表 2 中列出了部分规模的算法运行结果. 可以看到, 采用 EOX-GA 获得的调度方案其平均拖期数在各种规模下均优于文献 [5] 算法 LOX-GA 以及 EDD、MSLACK、CR 等并行机调度规则; 而且其运行速度明显优于 LOX-GA. 这主要得益于 EOX-GA 在编码和交叉方面的优势:

(1) 与 LOX-GA 相比, 本文所提的向量组编码方法具有以下明显优点:

- 编码简单;
- 存储空间节约: 染色体长度为  $n$ , 与加工机器数无关, 有利于计算机实现;
- 可行性: 编码过程直接满足工艺约束, 并为交叉和变异

提供了便利;

- 随机性: 基因  $g_i$  中  $j_i$  和  $k_i$  分别从工件集合与可行机集合中随机选择, 保证了加工路径和单机加工顺序的随机性;

- 快速性: 译码过程时间复杂度仅为  $O(n)$ , 有利于提高算法运行速度;

(2) 交叉过程自动满足可行性, 并较大程度保留了父代遗传信息, 进一步提高了寻优效果.

(3) 搜索过程一直在可行空间进行, 因此运行速度和寻优效果较好.

另外, 尽管算法比较过程并未对遗传参数进行特殊的调整, EOX-GA 仍获得了满意的结果, 这表明算法具有较强的鲁棒性.

#### 4.4 应用实例

为检验所提遗传算法的实际有效性, 利用某钢铁厂冷轧精整生产线的实际数据进行仿真. 该生产线属于典型的 C PSP 类型, 各机器可加工的钢卷类型受钢卷厚度限制(见表 3), 利用 EOX-GA 和文献[5]算法 LOX-GA 以及几种典型规则对该生产线  $20 \times 4$  规模求解(各算法均独立运行 20 次, 结果如表 4 所示), EOX-GA 结果令人满意.

表 3 4 台机器可加工的工件类型

机器	可加工钢卷类型				
M1	A	B	C	D	E
M2	A	B	C	D	
M3	A	B	C		
M4	A	B			

表 4 仿真结果

算法	最优值	平均值	最差值	标准差
EOX-GA	0	1.4	2	0.49
LOX-GA	2	2.7	3	0.51
EDD	4	4	4	0
MSLACK	3	3	3	0
CR	3	3	3	0

## 5 结束语

带工艺约束的并行机调度问题在生产线调度、并行计算和其它组合优化领域均有广泛应用前景, 遗传算法为 CSCP 提供了有效的求解途径. 本文所提的基于向量组编码的遗传算法较好的解决了最小化拖期任务数的 CSCP 问题, 大量仿真实验和实际生产数据的对比结果表明该算法具有较强的寻优能力和鲁棒性. 进一步工作可在处理其它调度目标以及结合启发式加快收敛性等方面展开.

## 参考文献:

[ 1 ] Gurses, Suerand E B. Minimizing the number of tardy jobs in identical machine scheduling [ J ]. Computers and Industrial Engineering, 1993, 25(4): 243- 246.

[ 2 ] Funda S S, Gündüz U. Parallel machine scheduling with earliness and tardiness penalties [ J ]. Computers & Operations Research, 1999, 26: 773- 787.

[ 3 ] 刘民, 吴澄, 戴元顺. 最小化拖期任务数并行机调度问题的一种基于知识的遗传算法 [ J ]. 电子学报, 1999, 27(9): 130- 132

[ 4 ] 攀登, 王安麟. 基于遗传算法的一类无序加工调度 [ J ]. 机械设计与研究, 1999, 15(1): 19- 21.

[ 5 ] 董斌, 李颢, 邵惠鹤, 等. 基于遗传算法的一类 Job shop 调度 [ J ]. 控制与决策, 1998, 13(1): 71- 74.

[ 6 ] 刘民. 遗传算法及其在生产调度问题中的应用研究 [ D ]. 博士学位论文. 北京: 清华大学自动化系, 1999.

[ 7 ] Goldberg D E, Ling R. Alleles, loci, and the travelling salesman problem [ A ]. Proceedings of the 9st International Conference on Genetic Algorithms and Their Applications [ C ], Grefenstette Erlbaum, 1985: 154- 159.

[ 8 ] Davis L. Applying adaptive algorithms to epistatic domains [ A ]. Proceedings of the 9th International Joint Conference on Artificial Intelligence [ C ], Grefenstette, 1985: 136- 140.

[ 9 ] Grefenstette J J. Optimization of control parameters for genetic algorithms [ J ]. IEEE Transactions on Systems, Man and Cybernetics, 1986, VSMC16(1): 122- 128.

## 作者简介:



尹文君 男. 1976 年生于江苏省泰兴市. 清华大学自动化系国家 CIMS 工程技术研究中心博士研究生, 目前感兴趣的研究领域为复杂制造系统智能优化理论与方法、进化计算、神经网络、免疫算法、人工生命等.

刘民 男. 1965 年生于浙江省杭州市. 清华大学自动化系国家 CIMS 工程技术研究中心讲师、博士, 中国自动化学会名词委员会副主任委员, 已在国内外刊物和会议上发表学术论文 20 多篇, 目前感兴趣的领域为复杂制造系统智能优化理论与方法、人工生命、进化计算等.