

一种求解柔性工作车间调度问题的混合遗传算法^①

宋莉波, 徐学军, 孙延明, 查 靛
(华南理工大学工商管理学院, 广州 510640)

摘要: 针对柔性工作车间调度问题(Flexible job-shop scheduling problem, FJSP), 提出了一种基于混合遗传算法的求解方案, 在初始种群中引入基于启发式规则生成的优良个体, 并使用有效的交叉、变异算子避免不可行个体的产生, 同时利用混沌序列的随机性和遍历性特点, 在遗传进化的过程中增加基于混沌序列的邻域搜索功能, 以提高遗传算法的执行效率. 通过仿真实验验证了该算法的可行性和有效性.

关键词: 柔性工作车间调度问题; 混沌; 混沌序列; 遗传算法

中图分类号: C934 **文献标识码:** A **文章编号:** 1007-9807(2010)11-0049-06

0 引言

柔性工作车间调度问题(Flexible job-shop scheduling problem, FJSP)是传统工作车间调度问题(Job-shop scheduling problem, JSP)的扩展, 突破了资源的唯一性约束, 每道工序的加工设备不再是唯一的, 而是一个可以选择的设备集合, 且该工序在每台设备上的加工时间是不同的. 与传统的 JSP 问题相比, FJSP 问题由于引进了加工工艺的柔性而更加符合生产实践, 同时却增加了问题的求解难度.

遗传算法(Genetic Algorithm, GA)是借鉴生物的自然选择和遗传进化机制而开发出的一种全局优化搜索算法. 理论上遗传算法依“概率 1”收敛于问题的最优解, 然而在实践应用中, 遗传算法会表现出早熟收敛和收敛速度过慢等缺点; 为了寻求一个问题的最优解, 算法有时不得不花费相当长的时间进行搜索, 尤其是在极值点附近, 个体之间非常相似, 搜索效率就更低了.

混沌是一种普遍存在的非线性现象, 具有随机性、遍历性、规律性和敏感性等特性. 其中, 混沌

的随机性和遍历性特点可以引入到智能算法中, 作为避免搜索过程陷入局部最小的一种优化机制^[8]. 本文提出了一种基于混合遗传算法的求解方案, 在遗传进化的过程中加入基于混沌序列的邻域搜索功能, 以提高遗传算法的执行效率. 通过仿真实验验证了该算法的可行性和有效性.

1 FJSP 问题描述及相关文献综述

与传统的 JSP 问题相同, FJSP 问题需要给出每道工序在某台设备上的加工起止时间, 但是 FJSP 比 JSP 更具挑战性的地方在于 FJSP 还需要为每道工序选择一个恰当的加工设备. 总的来说, FJSP 问题可以描述如下.

令 $J = \{J_1, J_2, \dots, J_n\}$ 表示所有的待加工的工件, 每个工件由 n_i 个工序组成, 表示为, $O_{i1}, O_{i2}, \dots, O_{in_i}$, 他们之间具有工艺的先后顺序. 令 $M = \{m_1, m_2, \dots, m_m\}$ 表示所有设备集合; 定义 M_j 为第 j 个工件的第 i 道工序 O_{ij} 的可用设备集合, 可见 $M_j \subseteq M$. 每道工序的加工时间随着所选设备的变化而变化, 令 p_{ijk} 表示工序 O_{ij} 在设备 m_k 上的加工

① 收稿日期: 2009-10-15; 修订日期: 2010-08-28.

基金项目: 国家自然科学基金资助项目(71071059); 国家自然科学基金资助项目(70872031); 教育部“新世纪优秀人才支持计划”资助项目(NCET-08-0212).

作者简介: 宋莉波(1975—), 女, 辽宁大连人, 博士生, Email: lhbosongtang@gmail.com.

时间. 调度的任务是在满足各种资源约束和工序前后关系约束的前提下, 把所有工序分配到可加工的设备上并确定工序的加工序列和开始加工时间, 使得调度的最大完工时间 C_{max} (即所有工件完工时间的最大值, *Makespan*) 达到最小. FJSP 包括两个子问题: ①为每道工序选择合适的设备, 即设备分配问题; ②在选定的设备集上安排所有工序的起止加工时间, 即工序的排列问题. 为了简化问题, 现提出以下假设: (1) 所有加工设备相互独立, 且在 $t = 0$ 时刻均为可用; (2) 所有工件相互独立, 且在 $t = 0$ 时刻均可被加工; (3) 工件的工艺计划是既定的, 即工序的先后顺序在加工过程中不能发生变化; 每个工序在可供选择的机器上的加工时间是确定的; (4) 每台设备一次只能加工一道工序, 同一道工序也不能在多台机床上同时加工; (5) 加工是非抢占式的, 即某个工序一旦开始加工就不能中断; (6) 不考虑 *Setup* 时间以及作业在不同机床之间的转移时间; 不考虑机床故障情况.

近年来, 学者使用不同方法对 FJSP 进行求解, 取得了一定的研究成果. 如 Mastrolilli 和 Gambardella 针对该问题提出了两种邻域函数, 并把这两种邻域函数集成到禁忌搜索算法中^[1]; F. Pezzella 和 G. Morganti 等使用遗传算法求解 FJSP 问题, 取得了较好的结果^[2]; Guohui Zhang 和 Yang Shi 等在使用遗传算法求解 FJSP 问题的同时, 加入了 Tabu 搜索寻找邻域解^[3]; 而 Jie Gao 和 Linyan Sun 等则提出了一种新的解决方法, 该方法综合利用了遗传算法的全局搜索能力和瓶颈移动法的局部搜索能力, 相比之下取得了更好的执行效果^[4]; 文献 [5] 和 [6] 则将文化算法融入到遗传算法中, 利用随机搜索中学习和进化的互动整合, 在进化过程中提取知识来指导下一步的搜索.

混沌是自然界的一种普遍现象, 有着精致的内在结构, 能在一定范围内按其自身规律不重复地遍历所有状态. 近年来, 不少学者将混沌理论与其它优化方法结合起来, 取得了较好的组合优化效果. 陈培友等^[7]设计了一种嵌入启发式规则的混沌搜索算法求解组合拍卖竞胜标确定问题, 克服了传统方法的不足, 取得了较好的执行效果; 杨波^[8]把混沌理论引入到遗传算法中, 利用混沌的

遍历性特点动态搜索个体的基因座空间, 进而改善了遗传算法的执行效率; 谷晓琳等^[9]为解决标准遗传算法收敛缓慢等缺点, 提出一种基于混沌变异算子的改进遗传算法, 把混沌运动的遍历范围“放大”到优化变量的取值范围, 通过多代进化求得问题的最优解; 匡胤等^[10]利用 Logistic 方程构造混沌算子, 形成混沌遗传算法, 通过实验对混沌算子和杂交算子、变异算子的性能进行了比较, 并将该算法应用在旅行商问题中进行性能检验, 结果表明算法的性能和稳定性都有较大提高; 文献 [11] 提出了一种混沌粒子群算法, 通过求解 FJSP 问题, 取得了一定的成果. 本文在以上研究成果的基础上, 提出一种基于混沌邻域搜索的遗传算法求解 FJSP 问题, 在遗传进化的过程中加入基于混沌序列的邻域搜索功能, 以提高遗传算法的执行效率.

2 求解 FJSP 问题的混合遗传算法

2.1 遗传算法架构

编码: 采用两段式编码, 即设备分配编码和工序排序编码. 为了直观说明编码的含义以及后续的交叉、变异、解码等操作, 现提出一个简单的示例, 包含两个工件和三台设备, 共计 6 个工序; 每个工序在可选设备上的加工时间见表 1, 其中“—”代表指定工序不能在相应设备上加工.

表 1 一个 P-FJSP 问题示例

Table 1 An instance of P-FJSP

工件	工序	设备 m_1	设备 m_2	设备 m_3
J_1	O_{11}	3	7	4
	O_{12}	6	—	2
	O_{13}	—	5	7
J_2	O_{21}	2	4	8
	O_{22}	9	1	—
	O_{23}	3	5	—

结合这个示例, 假如某个体的两段式编码如图 1 和图 2 所示. 图 1 为工序编码, 编码上方注明了每个基因对应的工序, 工序的排列体现了调度方案的优先关系, 即 $O_{11} > O_{21} > O_{12} > O_{13} > O_{22} > O_{23}$; 基因的数值是工件编号, 每个工件编号出现一次代表该工件的又一道工序得以安排. 图 2 为设备分配编码, 基因按照工序的定义顺序排列,

即从第一个工件的第一道工序开始, 到最后一个工件的最末一道工序结束. 从编码中可以看出, O_{11} 将在设备 m_1 上加工, O_{12} 将在设备上 m_3 加工, 以此类推.

O_{11}	O_{21}	O_{22}	O_{13}	O_{22}	O_{21}
1	2	1	1	2	2

图 1 基于工序的编码

Fig. 1 Machine assignment vector

O_{11}	O_{12}	O_{13}	O_{21}	O_{22}	O_{21}
1	3	2	1	2	2

图 2 设备分配编码

Fig. 2 Operation sequence vector

初始种群的生成: 为保证种群的多样性, 绝大部分初始种群个体是随机生成的. 在生成设备分配编码时, 需要从既定设备集合中随机挑选一个设备编号, 以保证个体的可行性. 为了加快搜索进程, 我们以小比率往初始种群中注入了基于启发式规则生成的优良个体. 其中, 设备分配所采用的启发式规则是“Approach by localization”^[12], 即优先分配加工时间短并且负荷低的设备给某个工序; 工序安排所采用的启发式规则是 SPT (Shortest Processing Time, 优先分配加工时间最短的工序) 和 MWKR (Most Work Remaining, 优先分配剩余工作量最多的工件).

解码: 解码操作即根据“设备分配编码”所指定的设备分配方案, 将所有工序按照“基于工序

的编码”所规定的先后顺序安排到对应设备上, 以确定每道工序的加工前后顺序. 具体做法是按顺序取出工序编码上的每个工序 O_{ij} , 在对应设备 m_k 上以最早允许加工时间加工; 即从零时刻到队尾时刻, 依次判断该机器上的各加工空闲时间段能否插入指定工序; 若能, 且加工开始时间晚于其紧前工序的结束时间, 则在空闲时间段中插入该工序; 否则, 以队尾时刻加工该工序, 即将此工序安排在当前队列的末尾.

适应度函数: 本文研究单目标 FJSP 问题, 目标是使得调度的最大完工时间 C_{max} (即 $Makepan$) 达到最小, 故令适应度函数 $fitness = M / C_{max}$, 其中 $M > 0$ 为一个较大的常数.

选择: 采用两个个体的锦标赛选择形式, 即在当前种群中随机选取两个个体, 选入适应度高的个体; 被淘汰的个体放回种群, 可以参加下一次的随机抽取.

交叉算子: 对于设备分配编码, 采用两点之间基因片段互换的交叉方式. 由于初始种群中所有个体的设备编码均是可行的, 故交叉后的个体将仍然保持可行; 而对于工序编码, 执行两点之间的基因段交换可能会产生不可行后代, 故本文采用两种交叉机制 GPX (Generalized partially mapped crossover) 和 GOX (Generalized order crossover)^[13] 执行个体工序编码的交叉操作, 其执行过程如图 3 所示.

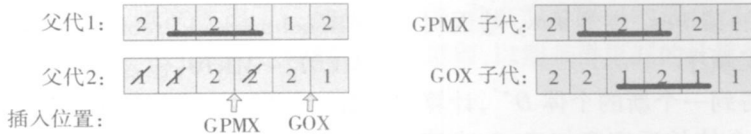


图 3 GPX交叉和 GOX交叉示例

Fig. 3 Instance of GPX and GOX crossover

第一步: 在父代 1 中随机选定一个基因片段“121”; 第二步: 按照基因片段“121”中每个基因值在父代 1 中的对应工序, 抹掉父代 2 中相应的基因值; 第三步: 把选定的基因片段插入到父代 2 中. 如果插入到原位置则为 GPX 交叉, 得到子代“2 121 21”; 如果插入到原基因片段后续第一个工序之前 (即第三个“1”之前), 则为 GOX 交叉, 得到子代“22 121 1”. 按照这两种机制生成的后代将保持可行, 不需再进行编码纠错.

变异算子: 设备分配编码的变异操作执行智能变异, 即尝试将父代个体中设备负荷最高的工

序移植到负荷较低的设备上执行; 工序分配编码的变异操作则随机选取一个工序, 将它安排到另外一个位置.

2.2 基于混沌序列的邻域搜索

2.2.1 混沌序列

利用一维或多维迭代映射的方法产生离散时间序列是研究分叉与混沌现象时最常用的手段之一. 定义下述方程

$$X_{n+1} = F(X_n), \quad n = 0, 1, 2, \dots \quad (1)$$

其中 X_n 是系统在 n 时刻的状态, $X_0 \in R^N$ 为系统

初值. 由此得到的时间序列 $\{X_n\}_{n=0}^{\infty}$ 称为系统的轨道. 在非线性迭代映射中, 只要参数选择满足一定条件, 任何阶的映射都可以产生混沌现象. 其中最常用的是 Logistic 映射, 又称抛物线映射, 是一维映射中最常用的映射, 其表达式为

$$X_i^{(n+1)} = \lambda X_i^{(n)} (1 - X_i^{(n)}), \quad 0 < \lambda < 4, \quad 0 \leq X_i^{(n)} \leq 1, \quad i = 1, 2, \dots, k \quad (2)$$

其中 λ 称为吸引子, 当 $\lambda = 4$ 时, 获得典型的 Logistic 映射:

$$X_i^{(n+1)} = 4X_i^{(n)} (1 - X_i^{(n)}), \quad 0 \leq X_i^{(n)} \leq 1, \quad i = 1, 2, \dots, k \quad (3)$$

其中 i 为混沌变量的序号, n 为迭代次数, 且 X_i^0 不能等于 0.025, 0.75, 1.

2.2.2 邻域搜索

为了加快最优解的搜索进程, 在遗传算法的每个进化代之中嵌入邻域搜索, 具体执行过程如下:

- 1) 假设当代的最优个体为 D^* , 对应的适应度值为 F^* , 基因长度为 L , 令 $n = 0$ 初始化 3 个混沌变量 X_1^0, X_2^0 和 X_3^0 , 设置邻域搜索步长为 N ;
- 2) 根据式 (3) 计算出下一代混沌变量 X_i^{n+1} , 并把它放大至能跟基因位相匹配的数字;

$Pos_{si} = \lceil X_i^{n+1} * L \rceil, i = 1, 2, 3$ 符号“ $\lceil \rceil$ ”表示向下取整.

a) 将设备分配编码的第 Pos_{s1} 位基因删除, 替换成一个不同的值; 即重新选择一个加工设备;

b) 将工序编码的第 Pos_{s2} 位和 Pos_{s3} 基因对调, 即调换它们的加工顺序.

3) 上述操作将得到一个新的个体 D^* , 计算它的适应度 f^* . 若新个体的适应度好于 f^* 或搜索次数已达到 N , 则搜索结束, 否则转到 2).

举个简单例子, 假设一个 FJSP 问题共包含 16 道工序, 在某代的最优个体如图 4 所示.



图 4 混沌映射示例: 旧个体

Fig. 4 Instance of chaotic mapping: old individual

三个混沌变量的初值为: $X_1^0 = 0.873, X_2^0 = 0.597, X_3^0 = 0.820$ 根据式 (3) 计算出下一代混沌

变量值为: $X_1^1 = 0.443, X_2^1 = 0.962, X_3^1 = 0.590$ 放大后的基因位为: $Pos_{s1} = 7, Pos_{s2} = 15, Pos_{s3} = 9$ 则把设备分配编码的 7 号位 (从 0 开始记位) 上对应设备号“3”替换成另一个设备; 选定新设备的方法可参照设备编码变异算子, 即选取负荷较低的设备替换现有设备, 假设我们经过比较选定设备“6”; 再把工序编码上第 9 位基因“2”和第 15 位基因“5”调换, 即 Job_2 的第 2 道工序和 Job_5 的第 3 道工序对调, 完成上述操作后的新个体如图 5 所示.



图 5 混沌映射示例: 新个体

Fig. 5 Instance of chaotic mapping: new individual

对这个新个体计算适应度的值, 并循环执行上述操作, 直至满足退出条件.

3 仿真实验

在 Pentium 4 CPU 主频 3.0GHz, 1GB 内存的 Window XP 操作系统下, 使用 Visual Studio 2005 C# 编程平台实现了本研究算法. 仿真实验所选取的数据是柔性作业车间调度问题的基准问题 (Benchmarked Problems) “Brandmarte” 数据中的 MK01 ~ MK10, 包含 10 ~ 20 个工作, 6 ~ 15 台设备, 工序数在 55 ~ 240 个之间; 实验的具体数据可以在网站 <http://www.idsia.ch/~monaldo> 上下载. 经过对本研究算法各参数的预实验, 最终设定种群规模 1000, 进化代数 200 代, 交叉概率 0.6, 变异概率 0.05, 邻域搜索步长为 50. 每个问题运行 20 次仿真实验, 记录所求得的最优值并计算平均值、平均运行时间和均值标准差, 各组实验的运行结果如表 2 所示.

“GENACE” 列是文献 [6] 的实验结果, “GA” 列是文献 [2] 的实验结果, “CGA” 列是本文所提出算法的实验统计结果, 其中 C_M^* 为本组实验最优值, $AV(C_M^*)$ 为实验均值, SD 为本组实验结果的标准差, $AV(CPU)$ 为本组实验的平均 CPU 时间. 表中带“*”号的数值表示该项实验结

果在同行中是最优的. 从表 2 可以看出, 本研究算法在 10 个问题上均取得了较好的计算结果, 其执行效果在总体上要好过文献 [2] 的单纯遗传算法和文献 [6] 的文化算法, 显示了本研究所提出算法的可行性和有效性; 中小规模问题的平均 CPU

时间在 10 秒以内, 大规模问题 (20 个工序) 的平均 CPU 时间不超过一分钟, 且每组实验结果的标准差均在 3% 之内, 其中 MK01、MK02、MK03 和 MK08 问题的 20 次运行结果为零方差, 显示了本研究算法的稳定性和鲁棒性.

表 2 仿真运算结果

Table 2 Computational results

问题	工件数 n	设备数 m	已知 LB	GENACE ^[6]	GA ^[2]	CGA 算法			
						C_M^*	AV(C_M^*)	SD(%)	AV(CPU)
M k01	10	6	36	41	40	40 [*]	40	0	0.5
M k02	10	6	24	29	26	26 [*]	26	0	1.8
M k03	15	8	204	204	204	204 [*]	204	0	8.9
M k04	15	8	48	67	60	60 [*]	60.6	0.40	1.7
M k05	15	4	168	176	173	173 [*]	174	2.49	6.5
M k06	10	15	33	68	63	58 [*]	62.7	1.33	6.0
M k07	20	5	133	148	139	142	142.5	1.69	12.3
M k08	20	10	523	523	523	523 [*]	523	0	7.3
M k09	20	10	299	328	311	307 [*]	309.8	0.98	19.8
M k10	20	15	165	231	212	201 [*]	212.7	1.40	39.3

4 结束语

遗传算法的优势在于全局寻优, 但是具有收敛速度慢、易于陷入局部最优的缺点. 而混沌序列的随机性和遍历性特点则给邻域搜索创造了很好的条件. 通过整合遗传算法的全局寻优能力和基于混沌序列的邻域搜索能力, 本文提出了一种求解 FJSP 问题的混合遗传算法, 在生成初始种群的时候注入少数由启发式规则生成的优良个体, 采用有效的交叉变异算子避免个体纠错, 在遗传进化的过程中执行混沌搜索, 使混合算法的全局搜索和邻域搜索达到适当平衡. 经仿真实验验证, 所提出算法是可行且有效的. 此外, 本研究算法的思路亦适用于其它离散组合优化问题. 下一步研究将在该算法中加入混沌交叉算子和混沌变异算子, 并在混合算法的进化过程中增加文化进化的思想, 以期获得更好的计算效果.

时候注入少数由启发式规则生成的优良个体, 采用有效的交叉变异算子避免个体纠错, 在遗传进化的过程中执行混沌搜索, 使混合算法的全局搜索和邻域搜索达到适当平衡. 经仿真实验验证, 所提出算法是可行且有效的. 此外, 本研究算法的思路亦适用于其它离散组合优化问题. 下一步研究将在该算法中加入混沌交叉算子和混沌变异算子, 并在混合算法的进化过程中增加文化进化的思想, 以期获得更好的计算效果.

参考文献:

- [1] Mastrolilli M, Gambardella LM. Effective neighborhood functions for the flexible job shop problem [J]. Journal of Scheduling, 2000, 3(1): 3-20
- [2] Pezzella G, Morganti G, Ciaschetti A. A genetic algorithm for the flexible job-shop scheduling problem [J]. Computers & Operations Research, 35 (2008): 3202-3212
- [3] Guohui Zhang, Yang Shi, Liang Gao. A genetic algorithm and tabu search for solving flexible job shop schedules [J]. 2008 International Symposium on Computational Intelligence and Design, 2008: 369-372
- [4] Jie Gao, Linyan Sun, Mitsuo Gen. A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems [J]. Computers & Operations Research, 35 (2008): 2892-2907
- [5] Nhu Binh Ho, Joc Cing Tay, Edmund M.-K. Lai. An effective architecture for learning and evolving flexible job-shop schedules [J]. European Journal of Operational Research, 179 (2007): 316-333
- [6] HO N B, Tay J C. GENACE: An efficient cultural algorithm for solving the flexible job-shop problem [C] // Proceedings of the Congress on Evolutionary Computation CEC, 2004: 1759-1766

- [7]陈培友, 汪定伟. 组合拍卖竞标确定问题的混沌搜索算法[J]. 管理科学学报, 2003(5): 24–28
Chen Peiyu Wang Dingwei Chaotic search algorithm for winner determination in combinatorial auctions[J]. Journal of Management Sciences in China 2003(5): 24–28 (in Chinese)
- [8]杨 波. 基于混沌理论的遗传算法改进及应用研究[D]. 南京: 南京理工大学, 2003: 26–44
Yang bo A study on the improvement and applications of Genetic algorithm based on chaotic theory[D]. Nanjing Nanjing University of Science 2003: 26–44 (in Chinese)
- [9]谷晓琳, 黄 明, 梁 旭. 混沌变异算子的改进遗传算法及其应用[J]. 计算机应用, 2007(10): 2490–2492
Gu Xiaolin Huang Ming Liang Xu Improved genetic algorithms based on chaotic mutation operation and its application [J]. Journal of Computer Applications 2007(10): 2490–2492 (in Chinese)
- [10]匡 胤, 刘益和, 黄迪明. 混沌理论在遗传算法中的应用研究[J]. 四川理工学院学报(自然科学版), 2007(8): 1–3
Kuang yin Liu yihé Huang diming Research on genetic algorithm based on chaos theory[J]. Journal of Sichuan University of Science & Engineering (Natural Science Edition), 2007(8): 1–3 (in Chinese)
- [11]谷 峰. 柔性作业车间调度中的优化算法研究[D]. 合肥: 中国科学技术大学, 2006: 66–73
Gu feng Research on optimization algorithms of flexible job shop scheduling problem [D]. Hefei China University of Technology 2006: 66–73 (in Chinese)
- [12]Kacem I Hammadi S Bome P. Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems[J]. IEEE Transactions on Systems, Man and Cybernetics, Part C 2002 32(1): 1–13
- [13]Bierwirth C, Mattfeld D and Kopfer H. On permutation representations for scheduling problems[C] // H M Voigt (Ed.) Proceedings of Parallel Problem Solving from Nature IV, Berlin, Germany Springer 1996 310–318

A hybrid genetic algorithm for flexible job shop scheduling problem

SONG Li-bo, XU Xue-jun, SUN Yan-ming, ZHA Jing

School of Business Administration, South China University of Technology, Guangzhou 510640, China

Abstract A genetic algorithm combined with local search is proposed to solve the FJSP with MAKESPAN criterion. A small percentage of elitist individuals are introduced into the initial population to fasten GA's convergence speed. Efficient crossover and mutation operators are adopted to avoid infeasible solutions and to hasten the emergency of optimum solution. During the local search process, Logistic chaotic sequence is adopted to explore better neighborhood solutions around the best individual of the current generation. Representative flexible job shop scheduling benchmark problems are solved in order to test the feasibility and validity of the proposed algorithm.

Key words flexible job-shop scheduling problem; chaos; chaotic sequences; GA