

并行流程车间调度问题及其概率学习进化算法

庞哈利, 万珊珊

(东北大学 信息科学与工程学院, 辽宁 沈阳 110004)

摘要: 并行 Flow shop 调度问题兼有并行机器和流程车间调度问题的特点, 是一类新型的调度问题. 针对最小化最大完工时间目标函数, 建立了一般并行 Flow shop 调度问题的整数规划模型. 鉴于问题的求解复杂性, 设计了基于概率学习的求解算法. 对随机生成的测试问题进行求解, 实验结果显示出该算法求解并行 Flow shop 调度问题的良好潜能.

关键词: 并行流程车间; 调度; 概率学习; 进化算法

中图分类号: TP13 **文献标识码:** A

Parallel flow shop scheduling problem using probability learning based evolutionary algorithm

PANG Ha-li, WAN Shan-shan

(School of Information Science & Engineering, Northeastern University, Shenyang Liaoning 110004, China)

Abstract: The parallel flow shop scheduling problem is a new kind of scheduling problem, which possesses the characteristics of both the parallel machines scheduling and the flow shop scheduling. A general parallel flow shop problem with the objective of minimizing makespan is addressed, and a mixed integer programming model is given. In view of the intractable nature of the problem, a probability learning based evolutionary algorithm is proposed to solve the problem. This algorithm is tested on some randomly generated problems. Computational results show that the proposed approach is accurate and effective.

Key words: parallel flow shop; scheduling; probability learning; evolutionary algorithm

1 引言(Introduction)

并行(Parallel) 机器和流程车间(Flow shop) 调度问题已得到广泛研究^[1~4], 而作为上述两类问题扩展的并行 Flow shop(Parallel flow shop) 调度问题却少有研究. 在电缆、钢丝绳和家电等行业的生产实际中, 经常存在多个并列的流程式加工过程, 每一加工过程的相应工序由型号相同或不同的机器构成, 但完成的加工功能是相同的. 这种由生产实际引出的并行 Flow shop 调度问题, 兼有并行机和 Flow shop 调度问题的特点, 构成一类新型而且特殊的多机器多阶段调度问题. 文献[5]研究了两条并行(并行度为 2) 的两阶段 Flow shop 调度问题, 考虑同一工件在各 Flow shop 的相应工序加工时间成比例的特殊情况, 给出整数规划模型和求解最小化最大完工时间(Makespan) 的启发式算法.

本文考虑更一般的情况: 加工环境由 K 个并行 Flow shop 构成(并行度为 K), 每个 Flow shop 包含

M 个机器(工序), 各 Flow shop 的机器构成没有固定的关系, 即工件在各 Flow shop 的相关工序的加工时间无统一的比例关系. 针对最小化最大完工时间调度目标函数, 给出混合整数规划数学模型, 并设计了基于概率学习的进化求解算法.

2 并行 flow shop 调度问题(Parallel flow shop scheduling problem)

并行 Flow shop 调度问题可描述如下: N 个工件在具有 M 台机器(工序) 的 K 条流程式生产环境(Flow shop) 中加工, 每一工件仅能在其中一条生产线上加工, 且须在此生产线上完成所有 M 道工序, 假设工件在各生产线上以排列排序(Permutation) 方式加工, 每台机器在同一时间仅能加工一个工件, 每个工件在同一时间仅能在一台机器上加工, 加工过程不允许中断, 要求确定每条生产线上的工件分配和加工顺序, 以使某一目标达到最优.

定义符号和变量如下:

J 为加工工件集合, $J = \{1, 2, \dots, N\}$; L 为并行度, 并行生产线(Flow shop)集合, $L = \{1, 2, \dots, K\}$; i, j 为工件编号, $i, j = 1, 2, \dots, N$; k 为并行生产线编号, $k = 1, 2, \dots, K$; m 为生产线机器编号, $m = 1, 2, \dots, M$; P_{ikm} 工件 i 在 k 生产线的 m 机器上的加工时间; C_{ikm} 为工件 i 在 k 生产线的 m 机器上的完工时间; S 为足够大的正数; $X_{ik} = 1$, 若工件 i 在 k 生产线上加工, 否则为0; $Y_{ijk} = 1$, 若在 k 生产线的 m 台机器上工件 i 领先于工件 j 加工, 否则为0.

并行 Flow shop 最小化最大完工时间调度问题数学模型为

$$\min \max_{i \in J} \left\{ \sum_{k=1}^K X_{ik} \cdot C_{ikM} \right\} \quad (1)$$

s. t.

$$\sum_{k=1}^K \sum_{i=1}^N X_{ik} = N, \quad (2)$$

$$\sum_{k=1}^K X_{ik} = 1, \quad i = 1, 2, \dots, N, \quad (3)$$

$$C_{jkm} - C_{ikm} + (3 - X_{ik} - X_{jk} - Y_{ijk}) \cdot S \geq P_{jkm}, \\ i, j = 1, 2, \dots, N; k = 1, 2, \dots, K; m = 1, 2, \dots, M, \quad (4)$$

$$C_{ikm} - C_{jkm} + (2 - X_{ik} - X_{jk} - Y_{ijk}) \cdot S \geq P_{ikm}, \\ i, j = 1, 2, \dots, N; k = 1, 2, \dots, K; m = 1, 2, \dots, M, \quad (5)$$

$$\left\{ \begin{array}{l} \sum_{k=1}^K X_{ik} (C_{ikm+1} - C_{ikm} - P_{ikm+1}) \geq 0, \\ i = 1, 2, \dots, N; m = 1, 2, \dots, M - 1, \end{array} \right. \quad (6)$$

$$X_{ik} = 1 \text{ 或 } 0, \quad (7)$$

$$Y_{ijk} = 1 \text{ 或 } 0. \quad (8)$$

并行 Flow shop 调度问题描述为混合整数规划模型. 式(1)为最小化最大完工时间目标函数; 约束(2)表示所有工件必须在各生产线上加工; 约束(3)表示每一工件仅能分配到一条生产线上加工; 式(4)和(5)表达了同一生产线上不同工件加工的顺序约束, 保证一台机器在同一时刻只能加工一个工件; 约束(6)描述了每一工件须依次在不同工序加工的顺序要求, 保证一个工件同时仅能在一台机器上被加工; 式(7)和(8)为决策变量取值约束.

3 基于概率学习的进化求解算法 (Probability learning based evolutionary algorithm)

3.1 PBIL 算法 (PBIL Algorithm)

PBIL (Population based incremental learning) 是一种基于概率分析的进化算法^[6]. PBIL 首先从一个均

匀分布的初始概率出发, 产生若干个随机的可行解—可行解种群, 通过对可行解种群中的个体的分析与评价获得知识, 根据学习所获知识修正生成概率, 进而指导后代的生成. 生成概率是整个进化过程信息的积累, 用它指导产生的后代将会表现出更优良的品性. PBIL 算法结构简单, 可以避免陷入局部最优, 在多个领域取得了较好的应用效果^[7~9]. PBIL 算法的基本流程如下:

初始化生成概率 P :

While (终止条件 = False)

{

通过生成概率 P 产生可行解种群;

计算种群中每个个体解的目标函数值;

确定种群中的最优个体解 B ;

根据最优解 B 修正生成概率 P ;

}

输出最优解;

应用 PBIL 算法涉及对问题空间的表达和算法的具体过程细节. 本文针对并行 Flow shop 调度问题的特点, 设计了相应的编码和算法过程, 构造了一种处理自然数与分隔符混合编码的新型的 PBIL 算法.

3.2 求解算法设计 (Algorithm design)

并行 Flow shop 调度问题求解归结为工件分配和排序两个相互关联的过程. 采用自然整数结合生产线分隔符编码方式, 同时刻画工件的分配与排序两方面情况, 可表达一个可行解的构成. 设 a_{kj} 表示 k 生产线上排在第 j 位的工件编号, 不同生产线以“*”分隔, 则总长为 $N + K - 1$ 的 K 段编码 $[a_{11} \dots a_{1j} \dots * a_{21} \dots a_{2j} \dots * a_{k1} \dots a_{kj} \dots]$ 可表达 N 个工件 K 条生产线 Flow shop 调度问题的可行解.

考虑到可行解中包含分隔符, 在个体解产生过程中, 分隔符与工件号一样对待, 其编号为 $N + 1$. 由此, 可行解编码中的每一基因位就有 $N + 1$ 个可能的取值, 即每一基因位有 $N + 1$ 个生成概率. 一个完整可行解的生成概率可以表达为 $(N + 1) \times (N + K - 1)$ 维的生成概率矩阵. 由生成概率产生个体解的过程采用转轮法, 即由同一基因位置的 $N + 1$ 个可能取值的生成概率相加, 得到 $N + 1$ 个累加概率, 产生一随机数, 累加概率中大于该数的最小累加概率所对应的取值即为该位置的基因取值. 设 s 为基因位, r 为基因位取值. 求解算法流程如下:

1) 初始化生成概率 $P_{rs} = 1/(N + 1)$; $r = 1, 2, \dots, N + 1, s = 1, 2, \dots, N + K - 1$;

2) 由生成概率随机产生规模为 Pop_size 的可

行解种群;

3) 计算每一可行解的目标函数值,并找出当前最优的解个体 $B = [b_1 b_2 \cdots b_l \cdots b_{N+K-1}]^T$;

4) 利用 B 修正生成概率 $P_{b_s} = P_{b_s} + \epsilon$ (这里 $s = 1, 2, \cdots, N + K - 1$, b_l 为 s 基因位所对应的基因位取值, ϵ 为概率修正系数);

5) 对每个基因位的生成概率进行归一化处理:

$$P_{rs} = P_{rs} / \sum_{r=1}^{N+1} P_{rs};$$

6) 若满足终止条件,输出求解结果,否则转 2)。

4 计算测试实验(Computational experiments)

为测试算法性能,对随机生成不同规模 $N \times M \times K$ ($N = 10, 20, 30, 40$; $M = 3, 4, 5, 7$; $K = 2, 3, 4$) 的并行 Flow shop 调度问题进行求解. 测试问题中工件的加工时间 P_{ikm} 为 $[1, 20]$ 之间的随机整数. 算法用 VC++ 6.0 编制,运行环境为主频 1 GHz 的个人计算机.

首先进行算法参数配置实验,以确定合适的算法参数. 然后考察算法的稳定性和准确性. 实验过程与结果如下:

1) 可行解种群规模和概率修正系数.

为便于比较和表示,引入相对目标值的概念. 设问题规模标识为 S , 算法参数(种群规模、概率修正系数)标识为 P , 计算目标值为 $f(S, P)$, 相对目标值 R_p 定义为: $R_p = \bar{f}(S, P) / \min_p \{f(S, P)\}$. 在不同的种群规模和概率修正系数下,对各组测试问题分别求解 10 次,测试种群规模和概率修正系数对算法性能的影响.

为了突出种群的作用,测试中算法的概率修正系数取值为 0.01, 算法终止条件为迭代次数 1000 代. 图 1 给出在不同种群规模下算法对三组问题的平均求解性能. 实验表明,在一定种群规模下(超过 50), 算法求解质量差异不大,即对种群规模敏感性不强,但种群规模对算法运行时间影响较大.

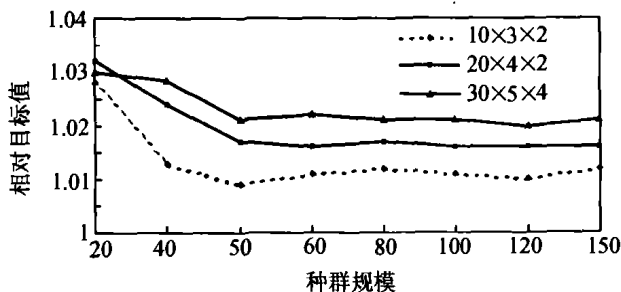


图 1 不同种群规模下算法性能

Fig. 1 Performance of the algorithm in different Pop_size

根据种群规模实验结果设置算法参数,考察不同概率修正系数对算法性能的影响. 图 2 为不同概率修正系数下,算法对三组问题计算的结果. 实验表明概率修正系数对算法的求解质量影响较大,对所讨论的问题概率修正系数取 0.02 较为理想.

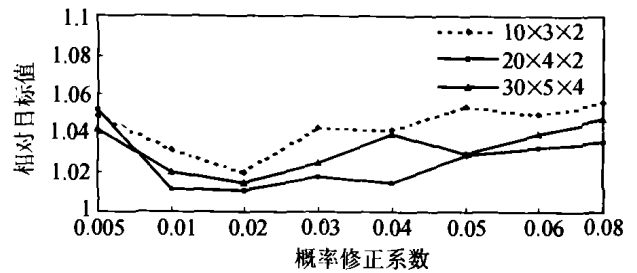


图 2 不同概率修正系数下算法性能

Fig. 2 Performance of the algorithm in different revising coefficient

2) 算法终止条件与收敛性能.

分别以熵、终止概率和最大迭代次数作为终止判据,考查算法的收敛性能. 熵可用来表示系统信息的一致性程度,在进化过程中,整个系统的熵值随着进化的进行不断减小. 熵的计算公式如下:

$$S_{entropy} = - \sum_{r=1}^{N+1} (\sum_{s=1}^{N+K-1} P_{rs} \ln P_{rs}). \quad (9)$$

采用终止概率作为算法终止条件,即当所有基因位的取值生成概率大于一个接近于 1 的正数时,算法终止. 这意味着此时各基因位的取值已趋向稳定,算法趋于收敛. 采用熵值和终止概率作为算法进化结束的判据,可比较准确反映算法的收敛程度,据此可控制算法的迭代次数和运行时间.

表 1 以 $20 \times 4 \times 3$ 规模问题为例给出算法在各种终止条件下的计算时间和平均目标函数. 另外从两个方面考察了算法的收敛情况:记录每代中 50 组解的平均值,得到算法的进化趋势如图 3 所示;记录每代最优解数量占种群规模总数的比例,考察此算法的收敛程度,收敛趋势由图 4 给出. 实验结果表明,基于概率学习的进化算法具有快速的进化性质和很好的收敛性能.

表 1 不同终止条件下算法的求解性能
Table 1 Results of the algorithm in different termination condition

结束条件	平均目标函数值	平均计算时间/s
迭代次数 1000	111.5	1.5
终止概率 0.99	113	2
熵值 0.001	112	4

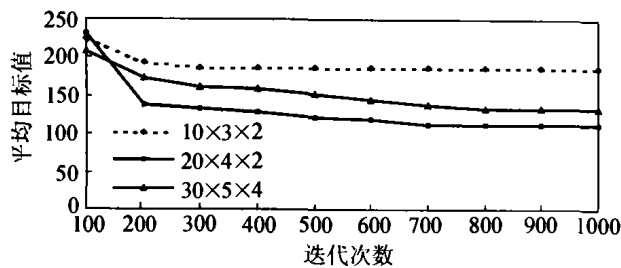


图3 算法进化趋势

Fig. 3 Evolutional trend of the algorithm

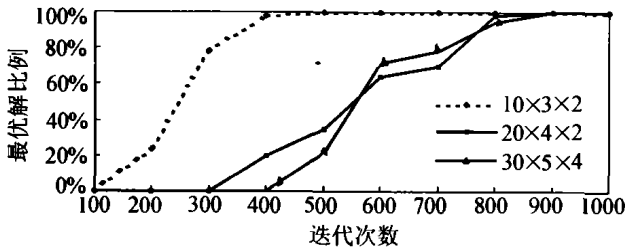


图4 算法收敛速度

Fig. 4 Convergent trend of the algorithm

3) 算法稳定性和计算准确性.

对4组问题,在相应算法最佳参数配置下进行实验测试,结果由表2给出.计算表明,对于不同规模的问题,算法求解的最大偏差小于4.5%,表现出稳定的求解性能.

表2 不同规模问题的求解偏差

Table 2 Deviations of solutions with different problem

测试问题	最优值	最差值	最大偏差	平均值	平均相对误差
10×3×2	186	190	0.021	187.5	0.008
10×5×3	81	83	0.024	81.5	0.006
20×4×3	111	114	0.027	111.6	0.005
20×5×3	129	132	0.023	129.6	0.005
30×4×3	149	151	0.013	150.2	0.010
30×5×4	133	139	0.045	136.3	0.025
40×7×4	188	196	0.043	191.6	0.019

为考察算法的求解质量,进一步进行了如下测试实验:即算法正常运行结束后,在只接受目标函数改善的前提下,强制执行10000次随机的基因互换操作局部搜索.参与交换的可以是工件,也可以是分隔符,这种基因互换操作对于解空间是可遍历的,实施足够多的互换操作可以保证算法跳出局部最优.表3以20×4×3规模问题为例给出进行强制局部搜索后与未进行局部搜索的10次求解结果的比较.可以看出,两种算法求解结果基本一致,表明PBIL算法得到的求解结果至少已极为接近问题的最优解,具有良好的寻优准确性.

表3 算法求解准确性比较

Table 3 Experiment results of algorithm in veracity

实验序号	1	2	3	4	5
基本算法	113	115	113	113	111
强制基因互换	113	115	112	113	111
实验序号	6	7	8	9	10
基本算法	113	115	113	114	111
强制基因互换	111	112	110	114	110

综合以上实验结果,本文提出的基于概率学习的进化算法可以很好地解决并行Flow shop问题.在合适选取算法参数前提下,可以快速获得稳定的、高质量的求解结果.

5 结束语(Conclusion)

针对实际的生产环境,提出一般并行Flow shop调度问题,并建立了整数规划模型,设计了基于概率学习的进化求解算法,通过计算实验研究了算法参数配置,测试了算法的收敛性能、计算时间、稳定性和准确性.计算实验结果显示出该算法求解并行Flow shop调度问题的良好性能.

参考文献(References):

- [1] DOFRAMACI A, SURKIS J. Evaluation of a heuristic for scheduling independent jobs on parallel identical processors [J]. *Management Science*, 1979, 25: 1208 - 1216.
- [2] LI C, CHENG T. The parallel machine minmax weighed absolute lateness scheduling problem [J]. *Naval Research Logistics*, 1993, 41: 33 - 46.
- [3] GAREY M R, JOHNSON D S, SETHI R R. The complexity of flow-shop and job-shop scheduling [J]. *Mathematics of Operations Research*, 1976, 1: 117 - 129.
- [4] WIDMER M, HERTZ A. A new heuristic method for the flow shop sequencing problem [J]. *European J of Operational Research*, 1989, 41: 183 - 193.
- [5] SUNDARAGHAVAN P, KUNNATHUR A, VISWANATHAN I. Minimizing makespan in parallel flow shop [J]. *Operational Research Society*, 1997, 48: 834 - 842.
- [6] BALUJA S. *Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning* [R]. Pittsburgh, PA, USA: School of Computer Science, Carnegie Mellon University, 1994.
- [7] SUKTHANKAR R, BALUJA S, HANCOCK J. Multiple adaptive agents for tactical driving [J]. *Applied Intelligence*, 1998, 9: 7 - 23.
- [8] SALUSTOWICZ R, SCHMIDHUBER J. Probabilistic incremental program evolution [J]. *Evolutionary Computation*, 1997, 5(2): 123 - 141.
- [9] 金丙尧, 蔚承建, 何振亚. 一种用于优化搜索的学习算法[J]. 软件学报, 2001, 12(3): 448 - 453.
(JIN Bingyao, WEI Chengjian, HE Zhenya. A learning algorithm for optimum search [J]. *J of Software*, 2001, 12(3): 448 - 453.)

作者简介:

庞哈利 (1963—), 男, 博士, 东北大学信息科学与工程学院教授, 研究方向为生产计划与调度、智能优化算法、ERP系统的建模与优化, E-mail: panghali@hotmail.com;

万珊珊 (1980—), 女, 学士, 东北大学信息科学与工程学院硕士研究生, 研究方向为进化算法的研究与应用.