第 26 卷第 4 期
1997 年 8 月

信 息 与 控 制
Information and Control

Vol. 26, No. 4
Aug., 1997

基于遗传算法的多机多阶段的 Flow Shop 问题

王 莉 王梦光
(东北大学信息科学与工程学院系统工程系 110006)

摘 要 讨论了多阶段多机的 Flow Shop 问题(Flow Shop with Multiple Processors 问题, 简称 FSMP 问题), 问题的目标是使工件的提前/拖期总成本最小. 这是一个 NP-难题. 为此, 首先给出了问题的数学模型, 然后构造了一个有效的遗传算法. 在本文的最后给出了实验结果和结论.

关键词 FSMP, 提前/拖期, 交货期, 遗传算法

1 引言

多阶段多机的 Flow Shop 问题(简称 FSMP 问题) 是一般的 Flow Shop 问题的推广, 它讨论的是 N 个工件在某些机器阶段具有并行机的流水车间的排序问题, 这个问题也可以看成是所有工件以同样的机器阶段顺序被加工的特殊的 Job Shop 问题. FSMP 环境相当普遍地存在于像钢铁工业, 化工业和石油工业等流程工业中, 1973 年, Salvador[1] 首先定义了 FSMP 调度问题. 之后, 在 20 多年中, 一些学者对 FSMP 问题进行了研究, 其中 Wittrock 在文献[2][3]中对 FSMP 排序问题分别给出了周期调度和非周期调度启发式方法, Salvadoor[1], Brah 和 Hunscke[5,6] 利用分枝定界法解决小规模的 FSMP 问题. 文献[7, 8]对两个阶段, 其中一个阶段有并行机, 另一个阶段是单机的特殊情况作了研究, 给出了近似解算法. 从文献上看, 所讨论问题的目标多数是最小化最大完工时间. 而对目标是提前/拖期的 FSMP 研究的却很少. 由于 FSMP 问题是 NP-难题[9], 因此, 用像分枝定界和动态规划等优化技术不能十分有效地解决该问题. 本文中, 我们用遗传算法求解提前/拖期的 FSMP 问题.

2 FSMP 问题的数学模型

2.1 FSMP 问题的描述

我们讨论的问题可以描述如下: 假设有 N 个工件, 每个工件都要依次经过 M 个机器阶段加工, 其中, M 个阶段中至少有一个阶段存在并行机. 从机器阶段的角度看, 所有工件的加工路线都是一致的. 所以, 此问题是一般的 Flow Shop 问题的推广. 在讨论中我们作如下的假设: (1) 每个工件在每个阶段且只能在其中一台机器上加工. (2) 工件在同一个阶段中的并行机上的加工时间相同. (3) 每台机器同时最多只能加工一个工件. (4) 工件一旦在一台机器上加工不允许中断. 问题的目标是找到一个调度以最小化工件的提前/拖期总成本.

2.2 FSMP 的数学模型

为了叙述方便, 我们引入下面的记号: N 为工件数; M 为机器阶段数; h_i, w_i 为工件 i 的单位提前/拖期成本; p_{ij} 为工件 i 在第 j 阶段的加工时间; d_i 为工件 i 的交货期; c_i 为工件 i 的完工时间; m_j 为第 j 阶段中的机器数; n^j_m 为分配在第 j 阶段的第 m 台机器上的工件数; e_{ij} 为工件 i 在第 j 阶段的完工时间; J_{jm} 为在第 j 阶段的第 m 台机器上的工件集合; t_{ij} 为第 i 工件在第 j

1997- 02- 19 收稿

个阶段的开始加工时间; $x_{ij}^{ms}=\begin{cases}1, & \text{如果工件 } i \text{ 在第 } j \text{ 阶段的第 } m \text{ 台机器上第 } s \text{ 个被加工} \\ 0, & \text{否则}\end{cases}$.

下面我们给出 FSMP 问题的数学模型

$$\min \sum_{i=1}^{N}(h_i\max\{0, d_i - c_i\} + w_i\max\{0, c_i - d_i\}) \quad (1)$$

st.

$$\sum_{s=1}^{n_m}\sum_{m=1}^{m_j} x_{ij}^{ms} = 1, \quad i = 1,2,\cdots,N;\ j = 1,2,\cdots,M \quad (2)$$

$$\sum_{i=1}^{N} x_{ij}^{ms} = 1, \quad j = 1,2,\cdots,M;\ m = 1,2,\cdots,m_j;\ s = 1,2,\cdots,n_m^j \quad (3)$$

$$t_{i,j-1} + p_{i,j-1} \leq t_{ij}, \quad i = 1,2,\cdots,N;j = 2,\cdots,M \quad (4)$$

$$\sum_{i_2=1}^{N} x_{i_2j}^{m,s+1} t_{i_2} - \sum_{i_1=1}^{N} x_{i_1j}^{ms}(t_{i_1} + p_{i_1j}) \geq 0, \quad j = 1,\cdots,M;m = 1,\cdots,m_j;s = 1,\cdots,n_m^j - 1 \quad (5)$$

$$\sum_{m=1}^{m_j} n_m^j = N, \quad j = 1,2,\cdots,M \quad (6)$$

$$x_{ij}^{ms} = 0, 1, \quad i = 1,2,\cdots,N;j = 1,2,\cdots,M;m = 1,2,\cdots,m_j;s = 1,2,\cdots,n_m^j \quad (7)$$

$$t_{ij} \geq 0, \quad i = 1,2,\cdots,N;j = 1,2,\cdots,M \quad (8)$$

其中 $c_i = t_{iN} + p_{iN}$.

模型中(1)式表示问题的目标是使工件的提前/拖期总成本最小;(2)式表示工件 i 在每个阶段中必须被加工一次;(3)式表示任一工件必在某一阶段中的某一机器上加工;(4)式表示同一工件必须在前一机器阶段完工后才能在下一机器阶段进行加工;(5)式表示在同一台机器上前一工件加工完之后才能加工后一工件;(6)式表示每个阶段中所有机器上加工的工件数之和等于工件总数;(7)和(8)式表示决策变量的取值范围.

3　FSMP 问题的遗传算法

3.1　遗传算法概述

遗传算法自从 1975 年由 Holland 提出至今已被广泛的应用到组合优化、机器学习、模式识别以及人工神经元网络等方面,它有 3 个基本算子:

(1) 选择: 从群体中为下一代遗传选取合适的个体,通常采用的选择方式是滚轮盘方式.

(2) 交叉: 按交叉概率从群体中随机地选取两个个体,以这两个个体为双亲,然后随机选取一个交换位置,两双亲相互交换各自的内容,从而产生两个新的个体,成为它们的下一代.

(3) 变异: 根据一定的变异率,在某个个体中随机选择一位,然后改变该位的值. 遗传算法的具体步骤可表述如下.

```
begin
{
    t= 0;
    initialize pop(t);
    evaluate each chromosome in pop(t);
    while terminal condition not satisfied do:
    begin
    {
```

```
        t= t+ 1;
        select pop(t) from pop(t- 1);
        cross;
        mutate;
        evaluate each chromosome in pop(t);
    }
    end
}
end
```

3.2 染色体的表示方法

设 $X = (a_{11}, a_{12}, \cdots, a_{1M}, a_{21}, a_{22}, \cdots, a_{2M}, \cdots, a_{N1}, a_{N2}\cdots a_{NM})$ 表示一个染色体. 其中 a_{ij} 表示工件 i 在第 j 阶段被分配的机器号. 由此得出 $J_{jm} = \{i \backslash a_{ij} = m\}$.

Peng 等[10] 给出了解不同交货期下的提前/提前调度问题的指数调度优先级函数

$$P_i(s_i) = \begin{cases} W_i, & s_i \leq 0 \\ W_i \exp[-((H_i + W_i)/H_i)(s_i/\bar{p})], & 0 \leq s_i \leq (W_i/(H_i + W_i))\bar{p} \\ H_i^{-2}[W_i - (H_i + W_i)s_i/\bar{p}]^3, & (W_i/(H_i + W_i))\bar{p} < s_i \leq \bar{p} \\ -H_i, & \text{否则} \end{cases} \tag{9}$$

其中, $W_i = w_i/p_i$, $H_i = h_i/p_i$, p_i 为工件 i 的加工时间; $s_i = d_i - t - p_i$, t 为机器最早可用时间; $\bar{p}$ 为 N 个工件的平均加工时间.

我们将(9)式扩展成并行机排序的优先级函数.

$i \in J_{jm} = \{i \backslash a_{ij} = m\}(j = 1, 2, \cdots, M; M = 1, 2, \cdots, m_j)$, 令 $s_{ij} = d_i - T_{ij} - p_{ij}$, $T_{ij} = \max(t, e_{ij})$, 其中, t 表示第 j 阶段第 m 台机器上最早可用时间.

下面我们给出工件 i 在第 j 阶段第 m 台机器上的优先级函数.

$$P_i(s_{ij}) = \begin{cases} W_{ij}, & s_i \leq 0 \\ W_{ij} \exp[-((H_{ij} + W_{ij})/H_{ij}(s_{ij}/\bar{p}], & 0 \leq s_{ij} \leq (W_{ij}/(H_{ij} + W_{ij}))\bar{p} \\ H_{ij}^{-2}[W_{ij} - (H_{ij} + W_{ij})s_{ij}/\bar{p}]^3, & (W_{ij}/(H_{ij} + W_{ij}))\bar{p} < s_{ij} \leq \bar{p} \\ -H_i, & \text{否则} \end{cases} \tag{10}$$

其中 $W_{ij} = w_i/p_{ij}$, $H_{ij} = h_i/p_{ij}$, $\bar{p} = (1/n_m^j)\sum_{i \in J_{jm}} \sum_{l=j}^{M} p_{il}$.

下面我们给出一个启发式将染色体映射成可行解.

(1) 设 $j = 1$.

(2) 在第 j 阶段中 $m = 1$.

(a) $J = J_{jm}$; $t = 0$.

(b) 运用(10)式计算出集合 J 中的每个工件的优先级, 从中选出工件 i_0 使

$$P_{i0}(s_{i_0,j}) = \max\{P_i(s_{ij}) \quad i \in J\} \tag{11}$$

如果有不止一个工件满足(11)式, 则任意选取其中一个工件.

(c) 将工件 i_0 排在第 m 台机器上加工的工件序列中的第一个可获得的位置上. 工件 i_0 在第 j 阶段上完工的时间是 $e_{ij} = \max\{t, e_{i,j-1}\} + p_{ij}$; $t = e_{ij}$.

(d) $J = J - \{i_0\}$; 如果 $J \neq$, 则转(b), 否则转(3).

(3) $m=m+1$; 如果 $m\leq m_j$, 则转(a); 否则转(4).

(4) $j=j+1$; 如果 $j\leq M$, 则转(2); 否则停止.

3.3 遗传算子

交叉算子: 从群体中按交叉概率随机地选取两个双亲 P_1, P_2. 随机地选取一个交叉位置, 然后进行简单交叉. 例如: 工件数是 10, 机器阶段数是 2, 第一阶段有 2 台机器, 第二阶段有 3 台机器. 设选取的两个双亲为

$$P_1=(1,3,1,1,2,1,2,\ \|\ 2,1,3,2,3,2,1,2,2,1,3,2,1)$$

$$P_2=(2,3,2,2,1,2,1,\ \|\ 1,2,1,1,3,2,1,2,3,1,2,2,3)$$

随机选择的交叉位置是 7("‖ "表示交叉位置), 交叉后得到下面的两个子代

$$O_1=(1,3,1,1,2,1,2,1,2,1,1,3,2,1,2,3,1,2,2,3)$$

$$O_2=(2,3,2,2,1,2,1,2,1,3,2,3,2,1,2,2,1,3,2,1)$$

变异算子: 按变异概率随机地选取一个染色体, 随机地选择 $1\sim N$ 之间的两个整数 q_1, q_2, 调换 $a_{q_1 1}, a_{q_1 2}, \cdots, a_{q_1 M}$ 和 $a_{q_2 1}, a_{q_2 2}, \cdots, a_{q_2 M}$ 的位置. 例如, 设选取的染色体为: $P_1=(1,3,1,1,\ \|\ 2,1\ \|,2,2,1,3,\ \|\ 2,3\ \|,2,1,2,2,1,3,2,1)$, 所选择的两个整数为 3 和 7, 经变异后得到下面的子代: $O_1=(1,3,1,1,2,3,2,2,1,3,2,1,2,1,2,2,1,3,2,1)$.

选择算子: 我们采取滚轮盘方式选择染色体, 以保证具有较高适应值的染色体在下一代中的后代个数也较多.

3.4 染色体的适应值

染色体的适应值为: fitness $=C_{max}-f$, 其中 f 是目标函数值, C_{max} 是足够大的正整数.

4 实验结果和结论

为了测试算法的有效性, 我们分别测试了下面 4 个不同规模的 FSMP 调度问题(见表 1) .

算法程序是用 MSC 语言编写, 在 PC 486 微机上运行. 由于初始种群是随机产生的, 所以对每个问题执行算法 20 次, 并设定种群的大小为 100, 交叉概率和变异概率分别为 0.9 和 0.2, 每次运行 150 代. 对每次运行记录最好目标值 Bf 和最坏目标值 Wf, 然后计算平均值 Mf. 设 $MB=(Mf-Bf)/Bf$; $WB=(Wf-Bf)/Bf$, 其实验结果见表 2.

表 **1**　问题的规模

问题	工件数	机器阶段数	各阶段机器数		
			1	2	3
P_1	10	2	2	3	
P_2	25	2	2	3	
P_3	50	3	2	3	3
P_4	100	3	3	5	2

表 **2**　实验结果

问题	Bf	Wf	Mf	WB(%)	MB(%)	CPU(s)
P_1	34	34	34	0.0	0.0	5
P_2	43	35	43.77	4.65	1.79	22
P_3	174	179	178.81	5.17	3.27	132
P_4	382	418	401.22	9.42	5.03	437

从实验结果看出最坏值和最好值的相对偏差不超过 10%, 平均值和最好值的相对偏差不超过 6%.

在这篇文章中, 我们讨论了目标是使工件的提前/ 拖期总成本最小的多阶段多机的 Flow Shop 问题, 给出了问题的数学模型, 并根据问题的特征构造了遗传算法. 实验结果表明此算法对 FSMP 问题是有效的.

参 考 文 献

1 Salvador M S. A Solution of a Special Class of Flow Shop Scheduling and Its Application. Springer-Verlag, Berlin, 1973

2 Wittrock R J. Scheduling Algorithms for Flexible Flow Line. IBM Journal of Research and Development, 1985, **29**(4): 401～412

3 Wittrock R J. An Adaptable Scheduling Algorithm for Flexible Flow Lines. Opers Res, 1988, **33**(4): 445～453

4 Kochhar S, Morris R J T. Heuristic Methods for Flexible Flow Line Scheduling. Journal of Manufacturing System, 1987, **6**(4): 299～314

5 Brah S A, Hunsucker J L. Optimal Scheduling in a Flow Shop with Multiple Processors. Paper Presented at the TIMS/ORSA Joint Nation Meeting in New Orleans, Louisiana, 1987, **5**: 4～6

6 Brah S A, Hunsucker J L. Branch and Bound Algorithm for a Flow Shop with Multiple Processors. Eur J Opl Res, 1991, **51**(1): 88～99

7 Gupta J N D. Two-stage Hybrid Flowshop Scheduling Problem. J Opl Res Soc, 1988, **39**(4): 359～364

8 Tsubone H. A Production Scheduling System for a Hybrid Flow Shop– a Case Study. OMEGA, Int J of Mgmt Sci, 1993, **21**(2): 205～214

9 Hunsucker J H, Shah J R. Performance of Priority Rules in a Due Date Flow Shop. OMEGA, Int J of Mgmt Sci, 1992, **20**(1): 73～89

10 Peng S O, Morton T E. The Single Machine Early/Tardy Problem. Management Science, 1989, **35**(2): 177～191

A GENETIC ALGORITHM FOR FLOW SHOP WITH MULTIPLE PROCESSORS

WANG Li WANG Mengguang

(*Department of System Engineering, Northeastern University, Shenyang* 110006)

Abstract This paper considers the scheduling problem of a flow shop with multiple processors at one or more stages(FSMP). The objective is to minimize the total earliness and tardiness costs of jobs. This is an NP-hard problem. We present a model of FSMP and construct a genetic assigning heuristic algorithm. Finally, experiment results are shown.

Key words FSMP, due date, earliness and tardiness, genetic algorithm

作者简介

王 莉, 女, 34 岁, 博士生. 研究领域为生产调度, 智能优化算法.

王梦光, 女, 61 岁, 教授, 博士生导师. 研究领域为大系统建模, CIMS 系统的设计与开发等.