文章编号: 1002-0411(2002)03-211-05

3 机 Flow-shop 调度问题研究

陈 雄[1,2] 汤光强[2] 吴启迪[2]

(1. 复旦大学电子工程系 上海 200433; 2. 同济大学 CIMS 研究中心, 上海, 200092)

摘 要: 提出了一种遗传分枝定界算法求解 3 机 Flow-shop 调度问题, 该算法类似于常用的遗传局部算法和遗传动态规划算法. 用随机方法生成测试例子, 通过与著名的 Taillard 的禁忌搜索算法和 Reeves 的遗传算法进行比较, 实验结果证实了遗传分枝定界算法的有效性. *

关键词: Flow shop 调度; 分枝定界; 遗传算法

中图分类号: TP13 文献标识码: B

STUDY OF THREE-MACHINE FLOW-SHOP SCHEDULING PROBLEM

CHEN Xiong TANG Guang-qiang[2] WU Qi-di

(1. *Department of Electronic Engineering, Fudan University* 200433; 2. *CIMS Research Center, Tongji University* 200092)

Abstract: A genetic branch and bound algorithm is proposed to solve the 3 machine flow-shop permutation problem, which is similar to often used the genetic local algorithm and the genetic dynamic programming algorithm. Based on the test examples that are produced with random methods and the performance Comparison with the famous Taillard' tuba search algorithm and Reeves' genetic algorithm, the experimental results show the effectiveness of the genetic branch and bound algorithm.

Keywords: Flow-shop scheduling, branch and bound algorithm, genetic algorithm

1 引言(Introduction)

Flow-shop 排序调度问题是指: n 个工件按同一的机器顺序在 m 台不同的机器上进行加工, 工件 i 在 m 机器上的加工时间为 t_{ij}(i= 1, …, n; j= 1…, m), 这些加工时间事先已知且固定不变, 其优化目标是最小化最后一个工件在最后一台机器上的完成时间(makespan). Flow-shop 调度问题存在以下假设: ①每个工件在每台机器上只能加工一次; ②每台机器一次只能加工一个工件; ③工件的加工不能发生中断.

文献[1]最早对 n 个工件两机问题进行了研究, 提出了工件最优排序的 Johnson 规则. 后来的研究人员在文献[1]研究的基础上, 开始探讨 $m \geqslant 3$ 的 Flow-shop 调度问题[2~3], 并证明了 Flow-shop 调度问题是 NP- 难问题($m \geqslant 3$)[4~4].

3 机 Flow-shop 调度问题的数学模型如下:

$$C(\sigma) = \max_{1 \leqslant x \leqslant y \leqslant n} (V_{xy}) \qquad (1)$$

目标是寻找一个 σ^* 使 $C(\sigma)$ 值最小. 这里, V_{xy} 为

$$V_{xy} = \sum_{j:\, j \mid\!\rightarrow x} t(j, m_1) + \sum_{\alpha=1}^{2} t(j, m_\alpha) + \sum_{j:\, x \mid\!\rightarrow j \mid\!\rightarrow y} t(j, m_2) + \sum_{\alpha=2}^{3} t(j, m_\alpha) + \sum_{j:\, y \mid\!\rightarrow J} t(j, m_3)$$

〗 其中: α= 1, 2, 3; σ 为 n 个工件按 m_1, m_2, m_3 的顺序在三台机器上进行加工的排序, 符号$\mid\!\rightarrow$ 表示如果 $i \mid\!\rightarrow j$, 则在当前排序 σ 中 i 排在 j 前.

针对该 3 机 Flow-shop 调度问题, 本文提出了一种新的遗传局部算法(简称遗传分枝定界算法), 即遗传算法中引入分枝定界算法保持对优化解有贡献的工件部分顺序, 求得 3 机 Flow-shop 调度问题的近优解. 为验证算法的有效性, 本文采用文献[6]的方法随机生成 3 机 Flow-shop 排序问题作为测试例子, 与目前最好的禁忌搜索算法[2]和遗传算法[3]两种改进算法进行比较, 大量的数值实验结果证实了本算法的有效性.

2 遗传分枝定界算法(Genetic branch-and-bound-algorithm)

文献[7-8]认为遗传算法[9]需与其他算法构成

* 收稿日期: 2000-12-28
基金项目: 国家自然科学资金(69774032)资助

遗传局部算法才能改善算法的寻优性能. 文献[10]构造了一种遗传动态规划算法求解排序问题, 其基本想法是应用动态规划保留问题解中的公有部分顺序, 求得问题的近优解. 本文的遗传分枝定界算法类似于文献[7- 8]的遗传局部算法和文献[10]的遗传动态规划算法, 在遗传算法中应用分枝定界算法保留 3 机调度问题解中的公共工件排序, 快速求得问题的近优解或最优解.

构造遗传分枝定界算法的思想源于文献[11]的研究结果, 即在两条最优或近优路径解中平均 85% 的两两城市排序是相同的, 而在 7 或 8 条最优或近优路径解中, 有 60% - 80% 的两两城市排序是相同的, 通过固定这种城市排序可以快速地求得旅行商问题的近优解. 遗传分枝定界算法的流程结构如下:

Step 1 (初始化): 产生 P 个初始解;

Step 2 (交叉和分枝定界改进): 重复以下 Step 2a、2b 和 2c, 直到获得 Q 个解;

Step 2a (交叉): 任选两个父代解, 并计算两个解中所共有的部分顺序 D;

Step 2b (变异): 随机地打乱所获得的部分顺序 D;

Step 2c (分枝定界算法): 对共有的部分顺序 D 应用分枝定界算法求得一个保留 D 顺序的最优解, 如果该解不在候选解集中, 就将该解添加到候选解集中;

Step 3 (选择): 从相应的 Q 个候选解中选取 P 个解;

Step 4 (终止): 重复 Step 2 和 Step 3 直到满足终止条件.

应用遗传分枝定界算法时有几点需要说明: ① P、Q($>P$)是事先设置的算法参数; ② 只有所有的 Q 个解都生成后, 才进行选择 P 个解; ③ 共有的部分顺序 D 作为方程(1)所描写的 3 机 Flow-shop 调度问题的工件顺序约束集, 这时, 原 3 机 Flow-shop 调度问题就成了具有工件优先顺序约束的 3 机 Flow-shop 调度问题. 应用分枝定界算法求得具有工件优先顺序约束的 3 机 Flow-shop 调度问题的最优解, 该解保留了部分顺序 D, 但不一定是方程(1)的最优解或近优解.

2.1 解个体的描述

本文直接采用 n 个工件的排序来表示 Flow-shop 调度问题的解个体, 即遗传染色体编码是 n 个工件一个可能的排序, 这是排序问题的一种自然表示法. 例如 X: 1, 2, …, i, …, N, 其中, 数 i 在 X 中的位置就是工件 i 在 flow-shop 调度问题中的加工排序, x 就是 flow-shop 调度问题的一个可行解.

2.2 交叉算子

基于排序表示的染色体编码方式, 传统的交叉算子无法保证产生的新个体仍然是合法的排序. 对于排序问题的自然表示法, 交叉算子可通过两种途径进行设计: 一种是对传统的交叉算子进行修改以满足子代个体的合法性[3]; 一种是直接设计满足排序染色体特点的交叉算子, 例如顺序交叉算子、位置交叉算子和边界交叉算子等. 通过实验比较顺序交叉算子、位置交叉算子和边界交叉算子的作用效果, 本文认为顺序交叉算子是最有效的. 顺序交叉算子是通过在一个父代排序上随机选择几个位置, 并将所选位置上的元素的顺序强加给另一个父代排序, 生成一个子代排序的过程.

2.3 变异算子

变异算子在遗传进化过程中主要作用是保证群体中个体的多样化, 防止算法过早地收敛于局部最优解. 通常, 变异算子随机地改变所选择的一个父代个体上的基因位置, 产生新的个体. 基于排序表示的染色体编码方式, 本文测试了顺序变异算子和转移变异算子的效果, 实验表明转移变异算子的效果更好, 因此采用转移变异算子实现打乱部分顺序 D 的工作, 产生具有新排序的个体. 转移变异算子的工作原理如下:

P_1: 2 1 4 5 3 7 6

* X

Q_1: 2 5 1 4 3 7 6

这里, "X"是随机地从部分顺序 D 中选取的元素, "*"是随机选取的左移或右移的位置数.

2.4 选择机制

对于最大值优化问题, 遗传选择机制通常采用个体适应值与群体平均适应值的比值来确定相应个体的生存概率, 适应值大的个体, 其生存机会必然就多. 赌盘选择就是这种典型的例子. 而对于最小值优化问题, 我们希望适应值小的个体的生存概率要大, 相反, 适应值大的个体的生存概率要小. 如果直接采用传统的选择机制, 就需要对个体的适应值进行某种修改, 否则, 就无法实现最小值优化问题的选择目的.

定义个体 p_n(n= 1, 2, …, P)的相对适应值函数 f_n 为:

$$f_n = U \min_{1\leqslant n\leqslant P} C_n + \max_{1\leqslant n\leqslant P} C_n - C_n \qquad (2)$$

这里, C_n 是个体 P_n 的目标值, U 是控制参数. 方程

(2) 中, 等式右边的第一项给出适应值的比例, 控制选择的情况, 第二项是常数, 保证适应值函数 f_n 非负.

$$P(n) = f_n/\overline{f} \tag{3}$$

这里, $\overline{f}=\frac{1}{P}\sum_{n=1}^{P} f_n$. 由方程(2)可以看出, 当 U 较大时, 即使最差的个体也最小的生存概率.

3　分枝定界算法描述(A describe of branch-and-bound algorithm)

本文应用分枝定界算法的目的是保留随机选出的两个父代个体中的公共部分排序, 以提高遗传算法的搜索性能. 分枝定界算法由约束选择的分枝过程和定界函数组成, 分枝和定界的确定决定了分枝定界算法的性能.

3.1　分枝描述

分枝过程是从搜索树上每个接点处构造其完整的排序开始的, 排序构造过程类似于拓扑排序过程.

构成 Flow-shop 问题的排序后, 可以通过寻找该排序的关键路径来减少 $C(\sigma)$ 值. 寻找关键路径的过程可以用寻找产生最大值 $C(\sigma)$ 的关键工件 x 和 y 来完成. 在关键路径中, 非关键工件 $j\in J-\{x, y\}$ 仅对一台机器贡献其加工时间, 如果该加工时间是工件 j 的最小值, 则 j 对该路径是最优的, 否则, 在满足约束关系的条件下, 移动的 j 位置可减少 $C(\sigma)$ 值. 设 j 相对于一个关键工件的优先关系约束为 r_j, 移动 j 所产生的值为 π, π_j 测定移动 j 引起当前关键路径的改变情况. 对于 3 机 Flow-shop 调度问题, 其关键路径只有两种类型, 一种是 V_{xx} 型, 另一种是 V_{xy} 型, 以下分别就这两种情况进行讨论.

3.1.1　关键路径具有 V_{xx} 型的情况

考虑两种情况获得工件集 Ω, 这里 $\Omega=\{i|$ 相对于关键工件 x, 工件 i 位置的颠倒能引起 $C(\sigma)$ 值的减少$\}$.

情况 **1**: 考虑 $j\in\{j|\ j\rightarrow x$ and $t(j, m_1) > t(j, m_3)\}$. 如果 j 排在 x 后而不违背当前的约束内容, 则 $\Omega:=\Omega\cup\{j\}$, 并令 $\pi_j=t(j, m_1)-t(j, m_3)$ 和 $r_j=x|\rightarrow j$.

情况 **2**: 考虑 $j\in\{j|\ x|\rightarrow j$ and $t(j, m_1) < t(j, m_3)\}$. 如果 j 排在 x 前而不违背当前的约束内容, 则: $\Omega:=\Omega\cup\{j\}$, 并令 $\pi_j=t(j, m_3)-t(j, m_1)$ 和 $r_j=j|\rightarrow x$.

3.1.2　关键路径具有 V_{xy} 型的情况

考虑三种情况获得工件集 Ω, 这里 $\Omega=\{i|$ 相对于关键工件 x 和 y, 工件 i 位置的转移能引起 $C(\sigma)$ 值的减少$\}$.

情况 **1**: 考虑 $j\in\{j|\ j|\rightarrow x\}$ 且它在机器 m_1 上的加工时间不是最小值 m_j. 如果 m_j 发生在 m_2 机器上, 且 j 可以排在 x 和 y 之间而不违背当前的约束内容, 则 $j\in\Omega, \pi_j=t(j, m_1)-t(j, m_2)$ 和 $r_j=x|\rightarrow j$. 如果 m_j 发生在 m_3 机器上, 且 j 可以排在 y 之后, 而不违背当前的约束内容, 则 $j\in\Omega, \pi_j=t(j, m_3)-t(j, m_1)$ 和 $r_j=y|\rightarrow j$.

情况 **2**: 考虑 $j\in\{j|\ x|\rightarrow j|\rightarrow y\}$ 且它在机器 m_2 上的加工时间不是最小值 m_j. 如果 m_j 发生在 m_1 机器上, 且 j 可以排在 x 前而不违背当前的约束内容, 则 $j\in\Omega, \pi_j=t(j, m_2)-t(j, m_1)$ 和 $r_j=j|\rightarrow x$. 如果 m_j 发生在 m_3 机器上, 且 j 可以排在 y 之后, 而不违背当前的约束内容, 则 $j\in\Omega, \pi_j=t(j, m_2)-t(j, m_3)$ 和 $r_j=y|\rightarrow j$. 如果约束内容不允许利用 m_j, 则如果 j 在其他机器上的加工时间 $t(j, m_1)$ 或 $t(j, m_3)$ 小于 $t(j, m_2)$, 可以按上述方式利用 j 在其他机器上的加工时间.

情况 **3**: 考虑 $j\in\{j|\ y|\rightarrow j\}$ 且它在机器 m_3 上的加工时间不是最小值 m_j. 如果 m_j 发生在 m_1 机器上, 且 j 可以排在 x 之前而不违背当前的约束内容, 则 $j\in\Omega, \pi_j=t(j, m_3)-t(j, m_1)$ 和 $r_j=j|\rightarrow x$. 如果 m_j 发生在 m_2 机器上, 且 j 可以排在 x 和 y 之间, 而不违背当前的约束内容, 则 $j\in\Omega, \pi_j=t(j, m_3)-t(j, m_2)$ 和 $r_j=x|\rightarrow j$. 如果约束内容不允许利用 m_j, 则如果 j 在其他机器上的加工时间 $t(j, m_1)$ 或 $t(j, m_2)$ 小于 $t(j, m_3)$, 可以按上述方式利用 j 在其他机器上的加工时间.

如集合 Ω 是空集, 则获得了当前约束内容下的最优排序, 否则, 从集合 Ω 中挑选具有最大值 π_j 的工件 j, 引进相应的约束 r_j, 这能极大地减少当前关键路径的值 $C(\sigma)$. 这样, 对于当前接点的两个子问题就可通过左分枝引入约束 r_j, 右分枝引入约束 $\overline{r_j}$ 而构成, 这里 $\overline{r_j}$ 是 r_j 的相反约束, 即, 如果 $r_j=j|\rightarrow x$, 则 $\overline{r_j}=x|\rightarrow j$. 这样产生的任意一个子问题的解就是具有排序约束关系的 3 机 Flow-shop 问题的解.

3.2　定界描述

将具有排序约束关系的 3 机 Flow-shop 排序问题构成三个 2 机 Flow-shop 排序问题进行考虑, 分别为: M_{12}, M_{23}和 M_{13}, M_{12}由机器 m_1和 m_2组成, M_{23}由机器 m_2 和 m_3 组成. 2 机 Flow-shop 排序问题可用文献[12]提出的方法得到有效地解决. 由文献[12, 13], 我们得到具有排序约束关系的 3 机 *Flow*-

shop 排序问题的三个低界, 为:

$$b_{12} = \delta_{12} + \min_j[t(j, m_3)],$$

$$b_{13} = \delta_{13} + \min_j[t(j, m_2)] \text{ 和 } b_{23} = \delta_{23} + \min_j[t(j, m_1)]$$

其中, δ_{12}, δ_{13}和 δ_{23}分别由文献[13]求得的 M_{12}, M_{13} 和 M_{23}的最优解. 因此, 具有排序约束关系的 3 机 Flow-shop 排序问题的低界为:

$$B = \max(b_{12}, b_{13}, b_{23})$$

有关这部分的详细内容, 有兴趣的读者请参阅文献[12, 13].

4 数值结果(Numerical result)

本文采用 Taillard 方法[6]产生 16 个 3 机 Flow-shop 排序问题, 工件数为 10 到 500 个, 以这些问题作为测试例子来验证遗传分枝定界算法的有效性. 在 Taillard 方法中, 伪随机数生成器基于回归公式: $X_{i+1} = (16807X_i) \bmod (2^{31} - 1)$, 该公式在闭区间[0, 2^{31}]产生一个均匀分布的数序, 这个数序按照

$$Z_i = a + \left[\frac{(b - a)X_i}{(2^{31} - 1)}\right]$$

生成闭区间 $U[a, b]$中的整数 Z_i, 本文取 $a = 1$, $b = 99$.

算法的性能是基于平均相对百分比偏差 γ 进行评价的:

$$\gamma = 100(C_{\max}(\sigma^H) - C_{\max}(\sigma^B))/C_{\max}(\sigma^B)$$

其中, σ^H 表示由算法 $H \in$ {Taillard 的禁忌搜索算法[2], Reeves 的遗传算法[3], 本文的遗传分枝定界算法} 求得的问题的排序, σ^B 是求得的已知最优解的排序.

三个算法的程序都用 C 语言编写, 在 PC586 (PII466) 上运行. 所有算法的终止条件采用文献[14]的终止条件, 即用相同的最大时间作为终止条件. 三种算法的初始解都采用 NEH[14]算法产生. 经过小规模问题的实验测试, 遗传分枝定界算法中的参数取为: 种群规模 $P = 30$, 候选解集中的个体数 $Q = 50$, $U = 0.01$, $p_c = 0.85$, $p_m = 0.5$. 每个算法对每个例子运行 10 次, 取 10 次结果的平均值进行比较. 实验结果如表 1 所示, 其中, 时间项是最早获得问题解的时间的平均值.

对于小规模问题, 如 $n \leqslant 40$ 时, 遗传分枝定界算法和 Taillard 的禁忌搜索算法都优于 Reesve 的遗传算法, 虽然在 $n = 10$ 时, 三种算法的寻优结果相同, 但遗传分枝定界算法获得最优解的平均时间最小, 禁忌搜索算法的(本节中的禁忌搜索算法是指 Taillard 的禁忌搜索算法) 时间次之, 遗传算法(本节中的遗传算法是指 Reesve 的遗传算法) 的时间最大. 对于大规模问题, 如 $n \geqslant 50$ 时, 遗传分枝定界算法和遗传算法无论在寻优性能上, 还是在获得最优解的平均时间上, 都优于禁忌搜索算法. 而遗传分枝定界算法与遗传算法相比, 显然, 遗传分枝定界算法的寻优性能强于遗传算法, 但在时间方面, 遗传分枝定界算法所用时间大于遗传算法的时间, 如 $n \geqslant 50$ 时, 除了 $n = 100$ 和 $n = 240$ 外, 其余的都是遗传分枝定界算法所用的平均时间较大.

表 1 三种算法的性能比较

Tab. 1 Performance comparison among the three algorithms

		遗传分枝定界算法		Reesve 遗传算法		Taillard 禁忌搜索	
n	m	γ	时间/s	γ	时间/s	γ	时间/s
10	3	0.00	6.51	0.00	8.62	0.00	7.12
20	3	0.00	14.76	0.01	15.12	0.00	12.86
30	3	0.05	21.29	0.19	24.58	0.11	28.62
40	3	0.23	41.42	1.17	38.31	0.54	42.17
50	3	0.00	49.22	1.48	43.67	1.65	75.12
80	3	0.36	63.96	1.11	58.76	2.36	98.35
100	3	0.12	57.63	4.54	79.63	5.17	121.76
160	3	1.28	112.76	2.69	99.78	4.10	146.81
200	3	2.21	153.48	4.13	115.34	5.06	192.69
240	3	0.31	138.35	3.79	165.89	3.96	253.15
280	3	1.53	181.41	6.18	143.92	6.91	248.98
300	3	0.00	251.23	10.61	186.77	12.85	283.37
350	3	0.54	271.85	3.24	243.58	8.79	321.78
400	3	1.15	296.54	5.52	254.31	6.73	331.64
460	3	0.16	342.92	8.11	311.15	8.43	355.45
500	3	1.02	321.33	5.87	287.11	9.82	415.87

5　结论(Conclusion)

针对 3 机 Flow-shop 排序问题, 基于文献[11]的研究结果, 本文提出了遗传分枝定界算法, 应用分枝定界算法保留问题解中的共有部分排序, 以有效地求得 3 机 Flow-shop 排序问题的近优解.

为验证遗传分枝定界算法的有效性, 本文应用文献[6]的方法随机生成测试例子, 与著名的 Taillard 禁忌搜索算法[2]和 Reeves[3]的遗传算法进行比较, 实验结果证实了遗传分枝定界算法具有很强的寻优能力. 其不足之处在于, 对于大规模问题, 遗传分枝定界算法与 Reeves 的遗传算法相比所用时间较大.

参　考　文　献 (References)

1　Johnson S M. Optimal two-and three-stage production schedules with setup times[J]. Naval Research Logistics Quarterly, 1954, (1): 61~ 68

2　Taillard E, Some efficient heuristic methods for the flow shop sequencing problem[J]. European Journal of Operational Research, 1990, (47): 65~ 74

3　Reeves C R. A genetic algorithm for flow shop sequencing[J]. Computers & Operations Research, 1995, (22): 5~ 13

4　Garey M R, Johnson D S, Sethi R. The complexity of flow shop and job shop scheduling[J]. Math Ops Res, 1976, (2): 117~ 129

5　Lenstra J K, Rinnooy Kan AHG and Brucker P. Complexity of machine scheduling problems[J]. Ann. Discrete Math, 1977, (1): 343~ 362

6　Taillard E, Benchmark for basic scheduling problems[J]. European Journal of Operational Research, 1993, (64): 278~ 285

7　Davis L. Job shop scheduling with genetic algorithms[A]. Grefenstette J. J. Proc. Int. Conf. Genetic Algorithms and Their Applications[C] Lawrence Erlbaum, 1985. 136~ 140

8　Whitley D, Starkweather T, Fuquary D. Scheduling problems and traveling salesman: the genetic edge recombination operator[A]. Schaffer J. D. Proc. 3rd Int. Conf. Genetic Algorithms[C]. Morgan Kauffmann. 1989, 133~ 140

9　Goldberg D E, Genetic algorithm in Search, optimization and machine learning[M]. Addison-Wesley, Reading, MA, 1989

10　Yagiura M, Ibaraki T, The use of dynamic programming in genetic algorithms for permutation problems[J]. European Journal of Operation Research, 1996, (92): 387~ 401

11　Lin S, Kernighan B W. An efficient heuristics algorithm for the Traveling Salesman Problem[J]. Operation Research, 1973, (21): 498~ 516

12　McMahon G B, Lim C J. The two-machine flow shop problem with arbitrary precedence relations, European Journal of Operational Research, 1993, **64**: 249~ 257

13　Lim C J, McMahon G B. The three-machine flow shop problem with arbitrary precedence relations, European Journal of Operational Research, 1994, **78**: 216~ 223

14　Nawaz M, Emscore Jr E E, Ham I. A heuristic algorithm for the-job,-machine flow shop arquebcing problem[J]. OMEGA, 1983, 11: 91~ 98

作者简介

陈　雄(1964-), 博士, 现为复旦大学电子工程系副教授. 研究领域为智能调度理论与应用、智能过程控制、非线形控制等.