文章编号: 1000-6788(2001)12-0066-06

求解作业排序问题的通用混合遗传算法研究

周　泓, 姬　彬
(北京航空航天大学经济管理学院, 北京 100083)

摘要: 车间作业排序理论是生产管理与组合优化领域的重要研究方向, 由于其固有的计算复杂性(NP-Hard), 一般无法利用经典方法求出最优解。本文针对一般作业排序问题, 将遗传算法与启发式方法相结合, 建立了一种混合算法框架, 利用遗传算法改进启发式方法的求解性能, 同时利用启发式方法引导遗传搜索过程, 以提高其搜索效率。通过对完工时间与平均延误时间等不同优化目标的计算分析与比较表明, 该方法对不同类型的排序问题均具有相当满意的求解效果。

关键词: 作业排序; 遗传算法; 启发式

中图分类号: O223; C931.1　　**文献标识码**: A

Study on the General Hybrid Genetic Algorithm for Job Shop Scheduling

ZHOU Hong, JI Bin
(School of Economics and Management, Beijing University of Aeronautics and Astronautics, Beijing 100083, China)

Abstract: Job shop scheduling is an important subject in the fields of production management and combinatorial optimization. It is usually hard to achieve the optimal solution with classical methods due to its high computational complexity (NP-Hard). A hybrid algorithm framework is proposed for general job shop scheduling problem in this paper, in which genetic algorithm (GA) is integrated with various heuristic methods. With this algorithm framework, the heuristics can be greatly improved by GA, while the searching efficiency of GA can be increased as well under the guidance of the heuristic rules. Finally, comprehensive numerical experiments have been made for optimizing makespan and mean tardiness, which show that satisfied solutions can be achieved for various scheduling problems with the hybrid algorithm.

Keywords: scheduling; genetic algorithm; heuristics

1　引言

车间作业排序(Job Shop Scheduling, 简记为JSS)是一个NP完全难题[1,2], 即使是工件与机器数量较少的简单问题, 要确切求出最优的加工顺序也是非常困难的。长期以来, JSS研究的方法大多以启发式方法为主, 尽管这些研究取得了一定的成果, 但是启发式方法本身却存在着难以克服的弱点, 如算法性能较差, 对问题的依赖程度高等等。近年发展起来的遗传算法(Genetic Algorithm, 简记为GA), 由于其固有的全局搜索特性, 被认为是一种切实有效的方法, 得到了日益广泛的研究。但该算法在求解JSS问题时也有其缺点, 例如, 随机的加工顺序组合会产生非常多的死锁情况, 从而大大降低了搜索效率。

已有研究表明[3,4], 将遗传算法与所求解问题的具体特点相结合, 对提高遗传算法的搜索效率起着重

收稿日期: 2000-03-30
资助项目: 国家自然科学基金(79970054); 航空基础科学基金(99J51068)

要作用。文献[5]利用这一思想将启发式方法与 GA 相结合, 建立了一种求解以完工时间(makespan)为优化目标的排序问题的混合算法, 利用遗传算法优化每台机器的首工序, 然后利用启发式方法安排每台机器的后续工序。本文在此基础上, 对算法进行了两方面的拓展, 一是引入机器的优先工序概念, 不仅利用 GA 优化每台机器的首工序, 而且优化每台机器的优先工序, 二是将算法推广至其它优化目标的排序问题, 即平均延误时间排序问题。

2　问题描述

本文所研究的 Job Shop 排序问题具有如下特点, 车间配备有 m 台机器, 将要加工 n 种工件, 每种工件的加工过程可由一系列工序组成, 每道工序由相应的机器完成, 每台机器同时只能加工一个工件, 而且机器在加工过程中不能被其它工件优先占用。本文研究了两种排序目标, 一是使所有零件的完工时间 $C_{\max}$ (makespan)最小, 二是使 n 个工件的平均延误时间 $\frac{1}{n}\sum_{j=1}^{n} T_j$ 最短, 其中, $C_{\max} = \max\{C_j\}$, $T_j = \mathrm{Max}\{0,\ C_j - d_j\}$, $j = 1, 2, \cdots, n$, C_j 为工件 j 的完工时间, d_j 为工件 j 的交货期。按文献[1]的分类, 这两类问题可被分别记作 $Jm \parallel C_{\max}$ 与 $Jm \parallel \frac{1}{n}\sum_{j=1}^{n} T_j$。

作业排序问题的解可以表示为每台机器上各工件加工顺序的排列, 即

$$
\begin{array}{lllll}
M_1: & O_{11} & O_{12} & \cdots & O_{1n} \\
M_2: & O_{21} & O_{22} & \cdots & O_{2n} \\
 & & \vdots & & \\
M_m: & O_{m1} & O_{m2} & \cdots & O_{mn}
\end{array}
$$

式中, O_{ij} 为机器 M_i 上的第 j 项加工操作, $O_{ij} \in \{J_1, J_2, \cdots, J_n\}$, M_i 表示第 i 台机器, J_j 表示第 j 个工件。

以上问题的解空间规模(所有可行解与不可行解)为 $(n!)^m$。显然, 即使 m 与 n 都不大(例如 10), 我们也无法遍历整个解空间去搜索最优解。

3　对问题 $Jm \parallel C_{\max}$ 求解的改进

文献[5]所设计的算法(GA-MWKR-NST), 对每台机器的第一道工序进行遗传搜索, 而对后续工序采取启发式调度规则(MWKR)来确定, 并利用邻域搜索法(NST)作局部辅助搜索。由于该算法对解空间作了过大的裁减(由 $(n!)^m$ 缩至 n^m), 因此不可避免地有可能遗漏更好的排序, 尽管增加了邻域搜索过程弥补, 收效却有限。本文在此基础上进行了改进, 即利用 GA 不仅优化首工序, 而且确定每台机器的优先工序所对应的零件。所谓优先工序, 是指只要该工序所对应的零件到达机器, 则机器加工完当前正在加工的零件后, 立即开工的工序, 故算法的基本过程为: 对每台机器的第一道工序和优先工序进行遗传搜索寻优, 然后按照启发式调度规则(MWKR)安排每台机器的后续工序, 以生成一个完整的加工顺序, 由此构成群体 $P(k)$, 并对 $P(k)$ 中的染色体进行复制、杂交、变异操作, 如此循环, 直至满足终止条件。我们简记该算法为 GA-QUE-MWKR。这种混合算法增加了对优先工序的选优, 以此取代邻域搜索过程, 可搜索到性能更好的解, 而且还可以大大减少“死锁排序”(与车间作业排序的基本约束条件不相容而无法实现的排序), 从而提高了算法的运算效率。此外, 本文设计的算法由于仅对每台机器的首工序和优先工序搜索寻优, 故保留了文献[5]算法减小搜索空间规模的优点。

3.1　问题编码

传统的遗传算法中, 最基本也最常用的染色体编码方式是利用二进制数 0 与 1 来表示基因。这种表示法在作业排序问题中显然是不合适的, 因为不仅复杂, 而且不易操作。最方便的编码方式是直接利用各机器上的作业排列来表示染色体, 如

$$O_{11} \quad O_{12} \quad \cdots \quad O_{1n} \quad O_{21} \quad O_{22} \quad \cdots O_{2n} \quad \cdots \quad O_{m1} \quad O_{m2} \quad \cdots \quad O_{mn}$$

式中, $O_{ij} \in \{J_1, J_2, \cdots, J_n\}$。每个染色体含有 m 个片段, 每段含 n 个基因, 分别表示 n 个工件在 m 台机器

上的投放顺序。本文设计的算法对每台机器的第一道工序和优先工序进行遗传寻优，第一道工序的染色体可表示为：

$$O_{11}O_{21}\cdots O_{m1}$$

优先工序的染色体可表示为：

$$Q_{11}Q_{21}\cdots Q_{m1}$$

式中，O_{ij}，$O_{ij}\in\{J_1, J_2, \cdots, J_n\}$。

为了便于遗传算子的操作，将上述两段染色体组合成新的染色体

$$O_{11}O_{21}\cdots O_{m1}\ \big|\ Q_{11}Q_{21}\cdots Q_{m1}$$

3.2 杂交算子

由于本文算法中染色体编码的特殊性，在此引入了成组杂交的技术，简述如下。对于 $m\times n$ 问题（m 台机器，n 个零件），以每台机器的首工序作为第一组基因片段，每台机器的优先工序作为第二组基因片段，进行分组：

$$O_{11}O_{21}\cdots O_{m1}\ \big|\ Q_{11}Q_{21}\cdots Q_{m1}$$

杂交时，针对不同的基因片段进行分区组合杂交。即在第一组基因片段中随机选取 i 至 j 位作为基因交换区，相应地第二组基因片段也要对应地选取 i 至 j 位作为基因交换区。实验表明，若两个基因片段不采用相同位置，则会引起性能下降，这是因为这样会破坏通过复制保留下来的良好模式。

例如，对于 6×6 的排序问题，将两个父代染色体分组，并选取第一组基因片段的 2 至 4 位作为基因交换区，则第二组基因片段的 2 至 4 位也相应地成为基因交换区（其中下划线部分即为基因交换区）。如下所示：

$$P_1:\ 1\ \underline{3\ 2\ 3}\ 4\ 6\ \big|\ 2\ \underline{4\ 5\ 1}\ 2\ 3$$

$$P_2:\ 2\ \underline{3\ 1\ 5}\ 1\ 4\ \big|\ 3\ \underline{1\ 6\ 1}\ 4\ 3$$

交换 P_1，P_2 的基因交换区的基因，得到下面两个子代

$$C_1:\ 1\ \underline{3\ 1\ 5}\ 4\ 6\ \big|\ 2\ \underline{1\ 6\ 1}\ 2\ 3$$

$$C_2:\ 2\ \underline{3\ 2\ 3}\ 1\ 4\ \big|\ 3\ \underline{4\ 5\ 1}\ 4\ 3$$

3.3 变异算子

本文设计了两种变异算子。第一种方法针对第一组基因片段，随机选取某些点，将当前零件号变为另一零件号。在上例中，对染色体 P_1，选取第一位、第四位进行变异，其操作如下：

$$P_1:\ \underline{1}\ 3\ 2\ \underline{3}\ 4\ 6\ \big|\ 2\ 4\ 5\ 1\ 2\ 3 \rightarrow \underline{5}\ 3\ 2\ \underline{2}\ 4\ 6\ \big|\ 2\ 4\ 5\ 1\ 2\ 3$$

在变异操作中一定要使变异后的染色体合法，为此必须注意在第一组基因片段选取的变异点，变异后的零件号不能与第二组基因片段对应位的零件号相同。例如在上例中：

$$P_1:\ \underline{1}\ 3\ 2\ \underline{3}\ 4\ 6\ \big|\ 2\ 4\ 5\ 1\ 2\ 3 \rightarrow \underline{2}\ 3\ 2\ \underline{2}\ 4\ 6\ \big|\ 2\ 4\ 5\ 1\ 2\ 3$$

即为非法变异，因为第一组基因片段中的第一位基因，其变异后与第二组基因片段的第一位基因相同，这使得零件 2 在机器 1 上加工了两次，故为非法排序。

第二种方法针对第二组基因片段，随机选取某些点进行变异。通过大量的算例分析，发现首工序对排序问题的影响比较大，故而本文以 0.7 的概率选取第一种变异方法，以 0.3 的概率选取第二种变异方法。

此外，本文的复制算子在选取个体时，是根据个体在群体中按适应值大小的排列顺序来加以选择的，适应值越高的个体，其序号越大，被选择的机会也越大。选择概率与个体的序号成比例（而不是与适应值大小成比例）。这样做的好处是可以大大简化适应值函数的构造，可以不受一些适应值函数特征（如非负）的约束。本文直接用目标函数值的负值作为适应值。

综上所述，本文的混合遗传算法可简要归纳如下

```
begin
    k: = 0;
    初始化 P(k);                /* P(k) 为第 k 代群体 */
```

求 $P(k)$ 中诸个体的适应值 $-C_{max}$;

while（不满足中止条件）do

begin

$k := k+1$;

{对 $P(k-1)$ 中的所有个体, 生成相应的由各台机器首工序和优先工序组成的染色体, 并由此构成集合 $P'(k-1)$; }

由 $P'(k-1)$ 通过复制生成 $P'(k)$;

在 $P'(k)$ 上实施杂交与变异操作;

{按照启发式规则($MWKR$)安排各台机器的后续工序, 以生成一个完整的排序, 并由此构成新的群体 $P(k)$; }

计算 $P(k)$ 中诸个体的适应值 $-C_{max}$;

end

end

3.4 算例分析

为了验证算法 GA-QUE-MWKR 的有效性, 本文应用文献[5]中的原始数据, 与单纯的遗传算法(GA)、文献[5]中所设计的算法(GA-MWKR-NST)进行了比较运算, 对于每种规模、参数的算例, 本文均运行了 20 次, 结果如表 1 所示。

显而易见, 算法 GA-QUE-MWKR 所求出的最好结果只有 1 例不如 GA, 其余全部优于 GA-MWKR-NST (其中有 1 例相等), 20 次运行的平均结果有 2 例不如 GA, 有 3 例不如算法 GA-MKWR-NST。正如算法设计时所预料的, GA-QUE-MWKR 的求解效率比 GA 要高得多, 但由于该算法增加了对每台机器优先工序的遗传搜索, 故其效率与 GA-MWKR-NST 相比略差, 以 10×15 规模为例, 在 PⅡ/333 微机上, 同样进化 3000 代, 算法 GA-MWKR-NST 平均运行 2 分 10 秒, 而算法 GA-QUE-MWKR 平均运行 2 分 40 秒, 然而该算法的求解性能与 GA-MWKR-NST 相比却得到了显著提高。

表 1　GA-QUE-MWKR 与其它算法的比较结果

算例 ($m\times n$)	GA		GA-MWKR-NST		GA-QUE-MWKR	
	最好	平均	最好	平均	最好	平均
10×10	3886	3906	3684	3743	3542	3647
10×10	2266	2274	2235	2284	2174	2215
10×10	3868	3931	3542	3569	3542	3591
10×10	2282	2352	2186	2271	2137	2186
10×15	3698	3734	3458	3543	3447	3535
10×15	3599	3640	3508	3611	3460	3541
15×15	4900	4900	4528	4887	4477	4831
15×15	4212	4280	4049	4241	4018	4322
15×20	5896	6077	5658	5868	5584	5741
20×20	6184	6297	6229	6844	6176	6970

4　对问题 $Jm \parallel \frac{1}{n}\sum_{j=1}^{n} T_j$ 的求解

通常, 一般形式的 Job Shop 问题均具有很高的算法复杂度, 而对同样的问题, 随着优化目标的不同, 算法复杂度也有差异, 现有 Job Shop 排序算法大多是使最大完工时间 C_{max} 达到最小, 若将优化目标换成平

均延误时间 $\frac{1}{n}\sum_{j=1}^{n}T_j$, 将大大增加算法的复杂性[1], 因此, 目前有关这类优化目标的排序问题研究大多局限于单机问题和并行机(parallel machines)问题(见文献[6- 9]), 很少能有效处理一般的多机Job Shop 问题。

本文将前面所建立的混合遗传算法框架应用于 $Jm \parallel \frac{1}{n}\sum_{j=1}^{n}T_j$ 问题的求解, 文献[8]通过大量的数值实验和广泛的比较分析, 认为启发式COVERT 在求解问题 $Jm \parallel \frac{1}{n}\sum_{j=1}^{n}T_j$ 问题时, 效果最好。因此, 本文利用 GA 对每台机器的首工序和优先工序进行搜索寻优, 然后利用COVERT 启发式安排后续工序。具体实现如下:

1) 由遗传进化过程(复制、杂交、变异)确定每台机器的首工序和优先工序;

2) 按 π_{ji} 值确定在机器 M_i 上后续各工件的加工优先级, π_{ji} 值大的工件优先加工。

$$\pi_{ji} = [1/p_{ji}] \cdot [1 - \max(0, d_j - l_{ji} - t)/(hl_{ji})]$$

式中, p_{ji} 为工件 j 在机器 M_i 上的加工时间, l_{ji} 为工件 j 在机器 M_i 上的加工提前期(lead time)的估计值, h 为可选参数(可取 h 为 1, 0.5 等)。

问题编码、算子设计、以及算法过程与上一节中求解 $Jm \parallel C_{\max}$ 问题一致, 只不过适应值由 $-C_{\max}$ 换为 $-\frac{1}{n}\sum_{j=1}^{n}T_j$, 在生成完整排序时, 启发式规则由MWKR 换成COVERT 而已。

表 2 $Jm \parallel \frac{1}{n}\sum_{j=1}^{n}T_j$ 排序问题的计算与比较

算例	规模 ($m \times n$)	紧迫交货期					宽松交货期				
		COVERT	GA-COVERT		相对改进率 (%)		COVERT	GA-COVERT		相对改进率 (%)	
			最优值	平均值	最优值	平均值		最优值	平均值	最优值	平均值
1	10×10	106.0	90.5	96.63	14.62	8.84	10.7	1.4	1.93	86.92	81.96
2	10×10	128.4	97.0	98.00	24.45	23.68	26.1	3.1	3.17	88.12	87.85
3	10×10	139.6	118.1	118.10	15.40	15.40	16.4	11.2	11.20	31.71	31.71
4	10×10	247.5	174.5	175.46	29.49	29.11	60.5	17.4	17.40	71.24	71.24
5	10×10	161.0	103.3	103.69	35.84	35.60	42.4	15.7	15.70	62.97	62.97
6	10×20	250.8	203.1	207.37	19.02	17.32	24.2	4.7	5.04	80.58	79.17
7	10×20	77.7	52.6	57.96	32.30	25.41	8.8	0.0	0.00	100	100
8	10×20	361.4	223.5	235.35	38.16	34.88	76.1	21.0	24.80	72.40	67.41
9	10×20	396.8	284.1	292.92	28.40	26.18	71.8	15.9	25.43	77.86	64.58
10	10×20	349.3	254.0	264.56	27.28	24.26	70.7	27.8	32.35	60.68	54.24
11	20×20	471.5	384.5	415.00	18.45	11.98	60.6	14.8	23.45	75.58	61.30
12	20×20	529.2	358.7	420.49	32.22	20.54	77.6	22.7	29.82	70.75	61.57
13	20×20	768.7	637.0	669.49	17.13	12.91	104.1	39.2	53.22	62.34	48.88
14	20×20	582.1	469.5	485.20	19.34	16.65	107.5	14.6	23.44	86.42	78.20
15	20×20	661.1	500.0	555.35	24.37	16.00	85.6	20.1	38.79	76.52	54.68

为检验算法性能, 本节随机生成了 15 个不同规模、不同参数的作业排序问题, 由于在含交货期的排序问题中, 交货期的紧迫与宽松程度对问题难度及求解性能有较大影响[10], 因而本文对每个算例均按交货期

的松紧分两种情况进行了考察。计算结果如表 2 所示, 其中,"COVERT"栏为单纯启发式方法所求得的结果, 而"GA-COVERT"栏则为本文混合遗传算法所求得的结果。对每个算例, 混合算法均做了 10 次独立运行, 每次的初始群体均随机产生, 每次运行进化 2000 代,"最优值"与"平均值"分别为 10 次运行的最佳和平均结果。

由表 2 可以看出, 不论问题规模大小, 也不论交货期紧迫还是宽松, 本文的混合算法(GA-COVERT)均比 COVERT 有较大的改进。除一个特例外(算例 7, 交货期宽松, 此时 GA-COVERT 求得的平均延迟时间为 0, 因而相对改进率为 100%), 最优值对 COVERT 的相对改进率最高可达 88.12%, 至少改进 14.62%; 在交货期宽松的情况下, 改进效果尤为显著, 几乎所有算例中(仅有一例除外), 最优值的相对改进率均超过 60%; 在所有算例中, GA-COVERT 算法 10 次运行的平均值对 COVERT 的改进也是相当可观的。因此, 本文所提出的 GA-COVERT 算法对求解排序问题 $Jm \parallel \frac{1}{n}\sum_{j=1}^{n} T_j$ 是非常有效的。

5 结束语

Job Shop 排序问题一直是生产管理领域研究的热点之一, 由于其固有的计算复杂性, 对于较大规模的问题, 很难找到一个有效的求解算法。现有方法大多局限于启发式方法, 其求解性能常常无法保证。遗传算法自问世以来, 虽然时间很短, 但已显示出了强大的生命力, 在众多领域得到了广泛应用。已有许多研究表明, GA 对 Job Shop 问题的有效求解具有极大的潜力。但由于遗传算法的固有特征, 其搜索效率尚有待进一步提高。本文将启发式方法与遗传算法相结合, 利用 GA 改进启发式方法的求解性能, 同时利用启发式规则引导 GA 的搜索过程, 以提高其求解效率, 并将这一混合算法用于求解 Job Shop 排序问题中的著名难题 $Jm \parallel C_{\max}$ 与 $Jm \parallel \frac{1}{n}\sum_{j=1}^{n} T_j$, 均取得了很好的效果, 表明这种算法结构对不同排序问题有较强的通用性。

参考文献:

[1] Pinedo M. Scheduling: Theory, Algorithms and Systems[M]. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[2] Rinnooy Kan A H G. Machine Scheduling Problems: Classification, Complexity and Computations[M]. The Hague: Martinus Nijhoff, 1976.

[3] Goldberg D E. Genetic Algorithms in Search, Optimization and Machine Learning[M]. New York: Addison-Wesley, 1989.

[4] Davis L. Handbook of Genetic Algorithms[M]. New York: Van Nostrand Reinhold, 1991.

[5] 周泓, 冯允成. 一种启发式混合遗传算法及其在车间作业排序问题中的应用[J]. 航空学报, 1998, 19(1): 74- 77.

[6] Chen C, Bulfin R L. Scheduling a single machine to minimize two criteria: maximum tardiness and number of tardy jobs[J]. IIE Transactions, 1994, 26(5): 76- 84.

[7] Davis J S, Kanet J J. Single machine scheduling with early and tardy completion costs[J]. Naval Research Logistics, 1993, 40: 85- 101.

[8] Morton T E, Pentico D W. Heuristic Scheduling Systems[M]. Chichester: John Wiley & Sons, 1993.

[9] Brucker P. Scheduling Algorithms[M]. 2nd edition. Berlin: Springer, 1998.

[10] Baker K R. Introduction to Sequencing and Scheduling[M]. New York: Wiley, 1974.