

车间作业调度中矩阵耦合问题求解

A Method of Solving Matrix Coupling in Job Shop Scheduling

大连民族学院 王万雷
大连理工大学 CIMS 中心 刘晓冰 邢英杰
大连海事大学 袁长峰

[摘要] 针对使用普通编码方式求解车间作业调度问题时出现的矩阵耦合问题,提出了一种基于遗传算法的新型编码方式。采用无延迟作业调度方法,不仅能避免死锁现象,而且显著提高了遗传迭代的速率。最终,仿真结果证明了本调度算法的有效性。

关键词: 遗传算法 编码 调度 矩阵耦合

[ABSTRACT] Aiming at solving matrix coupling problem that often appears in job shop scheduling with the ordinary coding mode, a new method of coding based on genetic algorithm which can resolve this problem is proposed. A scheduling method without postponement is applied, which not only keep from bad-scheduling happening but also promote genetic iteration speed to a great extent. Finally, simulation results demonstrate the effectiveness of the proposed method.

Keywords: Genetic algorithm Coding Scheduling Matrix coupling

近年来,基于群体进化的遗传算法以其独特的优化机理引起了众多研究者的兴趣,遗传算法已广泛应用于控制系统的参数优化、机器学习以及神经网络权值的学习和结构的优化等方面,为求解车间作业调度这类复杂问题提供了一条新的途径^[1]。

基于遗传算法的车间作业调度问题的求解,虽然国内外已有研究,但是在遗传算法的应用过程中,研究者一般采用的是间接的 0/1 编码方式,很少采用直接方式编码,这样做必须解决遗传过程中出现的矩阵耦合问题,增加了计算过程的复杂性。本课题提出的一种直接字符型编码方式,成功地解决了求解过程中出现的矩阵耦合问题。在作业调度过程中采用了无延迟作业调度方法,不仅解决了工件调度过程中出现的死锁现象,而且显著地提高了遗传迭代的速率。仿真结果证明了该方法的有效性。

1 问题的描述

车间作业调度问题的求解目标是: n 个工件在 m

台机器上加工,各工件的加工路线不同,如何安排每台机器上工件的加工顺序,在满足不超期交货的条件下,使得 n 个工件总的完工时间为最小^[2-3]。描述一个调度需要用 3 个参数,分别为 i, j, k 。其中 i 表示工件代号, j 表示工序号, k 表示完成工件 i 的第 j 道工序的机器的代号。因此,可以用 (i, j, k) 来表示工件 i 的第 j 道工序是在机器 k 上进行的。如果我们用 JM 来表示加工工序矩阵,用 $T(JM)$ 来表示该排列下的完工时间,那么我们的目标函数就是求矩阵 JM_0 使得 $T(JM_0)=\min T(JM)$ 。

2 建立数学模型

车间作业调度问题不仅具有极值性,即其结果是解值的最小化;而且具有有效性,即其结果必须符合车间作业调度问题的基本约束条件。车间作业调度问题的基本约束条件为:

- (1) 每道工序不能同时在几台不同的机器上加工,即每道工序必须在指定的机器上加工;
- (2) 每道工序必须在它前面的工序加工完毕后才能进入加工(如果一个工序在其之前的工序没有加工完毕就进入加工过程,即称之为死锁);
- (3) 同一时刻同一台机器只能加工一个零件,不允许同时加工多个零件。

以上是进行调度理论研究的主要约束条件,而能否有效地解决这些约束条件以满足要求则是调度理论研究的关键。

为了建立数学模型,先给出以下定义^[4]:

定义 1: 设 P 为 n 个工件的集合,则 P 可以表示为 $P=\{P_1, P_2, \dots, P_n\}$ 。

定义 2: 设 M 代表 m 台机器的集合,则 M 可以表示为 $M=\{M_1, M_2, \dots, M_m\}$ 。

定义 3: 设 P_i 的工序数目为 k_i , JP_i 代表 P_i 工件的工序集合,则 JP_i 可以表示为 $JP_i=\{J_i(1), J_i(2), \dots, J_i(k)\}$ 。

定义 4: 设机器 M_i 上的加工工序排列为 JM_i , 其中, $0 < i < m$, 则 JM_i 可以表示为 $JM_i=\{j(1), j(2), \dots,$

$j(l_i)$ }, 其中 l_i 为在机器 M_i 上加工工序的总数, $j(i)$ 为在机器 M 上加工的第 i 个零件的工序代号。

定义 5: 对定义 4, 如果设 $w = \max\{l_1, l_2, \dots, l_m\}$, 则可以用一个矩阵 JM 表示所有机器的工序排列, 我们称 JM 为机器加工工序排列阵, 记为

$$JM = \begin{bmatrix} 1(1) & 1(2) & \dots & 1(w) \\ 2(1) & 2(2) & \dots & 2(w) \\ \vdots & \vdots & \ddots & \vdots \\ m(1) & m(2) & \dots & m(w) \end{bmatrix},$$

$$\text{或者 } JM = \begin{bmatrix} JM_1 \\ JM_2 \\ \vdots \\ JM_m \end{bmatrix}$$

式中, JM_i 代表第 i 台机器上的加工工序排列阵。

定义 6: 有了机器加工工序排列阵 JM 就可以描述一个加工, 但还缺少一个描述加工时间的矩阵, 这里定义加工时间阵 T :

$$T = \begin{bmatrix} t_1(1) & t_1(2) & \dots & t_1(w) \\ t_2(1) & t_2(2) & \dots & t_2(w) \\ \vdots & \vdots & \ddots & \vdots \\ t_m(1) & t_m(2) & \dots & t_m(w) \end{bmatrix}。$$

该矩阵中的每一个元素都与矩阵 JM 中的元素一一对应。 $t_j(w)$ 的值表示 $j(w)$ 的完成时间。

基于上述定义, 现在可以给出车间作业调度问题数学模型的定义。

定义 7: 设 P 为工件集合, M 为机器集合, JM 为机器加工工序排列阵, T 为机器加工时间阵。如果对于一个给定的机器工序排列阵 JM^* , 满足如下关系式 $T(JM^*) = \min T(JM)$, 即 JM^* 使目标函数 $F(JM)$ 取值最小, 且与 T 相容, 则称 JM^* 为车间作业调度问题的最优解, 并称求解 JM^* 的过程为车间作业调度。这就是给出的车间作业调度问题模型的定义。其中目标函数为

$$\min[T(JM)] = \min\{\max[C(1), C(2), \dots, C(i), \dots, C(m)]\},$$

式中, $C(i)$ 为第 i 台机器的总加工时间, $T(JM) = \max[C(1), C(2), \dots, C(i), \dots, C(m)]$ 为完工时间。

3 编码^[5]

如果按机器加工工序排列阵 JM 和加工时间阵 T 分别作为染色体进行编码, 则能做到编码简单、运算效率又高; 同时也必须考虑矩阵 T 和矩阵 JM 的耦合问题, 即必须处理矩阵 T 和矩阵 JM 的相互关联性问题。这样做无疑增大了运算难度。

在这里, 我们采用单一编码方式, 将所有工件的工序统一编号。例如, 对于 n 个工件 m 台机器, 第 i 个工件的工序数为 k_i , $k = \max\{k_1, k_2, \dots, k_n\}$ 这样的调度问题可以以如下方式表示:

$$J[nk] = \{(1, 1, M_1, t_1), (1, 2, M_2, t_2), \dots, (n, k, M_m, t_{nk})\},$$

式中, $(1, 1, M_1, t_1)$ 表示第一个工件的第一个工序在机床 M_1 上加工, 加工时间为 t_1 ; (n, k, M_m, t_{nk}) 表示第 n 个工件的第 k 个工序在机床 M_m 上加工, 加工时间为 t_{nk} , 依次类推。其中小括号内的元素作为染色体的一个基因, 它是一个整体, 在遗传运算过程中不可分割。

在程序运行过程中, 把一个基因作为一个结构体来定义:

```
struct job
{
    int number;
    /* 定义工件的编号 */
    int pro_no;
    /* 定义该工件的工序编号 */
    int mach_no;
    /* 定义加工该工序的机床编号 */
    int time;
    /* 定义该机床加工该工序所需要的时间 */
}
```

这样定义后, 程序在运行过程中能够有效地保证该基因的有效独立性, 从而解决了矩阵 T 和矩阵 JM 的耦合问题。

4 调度结构

在调度问题上, 采用无延迟作业计划技术^[6]。对于多数实际目标函数来说, 根据无延迟作业计划生成的调度, 简称无延迟调度, 是一类非常重要的调度。生成无延迟调度比生成任意主动调度要容易。虽然无延迟调度不一定是最优的, 但大量试验结果表明, 在多数情况下它们比其他主动调度的效果好。因此, 在遗传算法中采用无延迟调度技术能使每一代个体表示的解都是近优解, 从而加快了迭代过程的收敛速度。

可安排操作集是调度生成中的一个有用概念, 简称可安排子集。在任一时刻, 可安排子集 S_w 是那些在其前面的操作已被排好的所有操作。对于一个 n 作业 m 设备问题, S_w 最初包括每个作业的第一个操作; 一段时间之后, S_w 的子集 S 被选中并移入调度。这时, 如

果存在的话,紧随 S 中每个操作之后的那些操作即被加入 S_n 中。这一过程重复进行,直到 S_n 成为空集,调度即完成。需要注意的是,当没有操作要在设备 M 上执行时,设备 M 即闲置。设备 M 的可安排子集可定义为分配给设备 M 的操作中那些其前面的操作已被排好的操作集。这样,我们得到的调度就可以很容易地避免工件的死锁现象^[6]。调度过程的流程图见图 1。

5 数据仿真

用不同规模的车间作业调度问题对上面提出的调度方法进行仿真,并将其结果与启发式算法 FCFS (First Come First Serve) 调度方法^[7]所得出的结果进行比较,见表 1。比较结果证明了该算法的有效性。

6 结束语

本课题采用了字符型直接编码方式,成功地解决了遗传算法应用于车间作业调度问题中产生的矩阵耦合问题。在工件调度过程中,应用无延迟作业调度

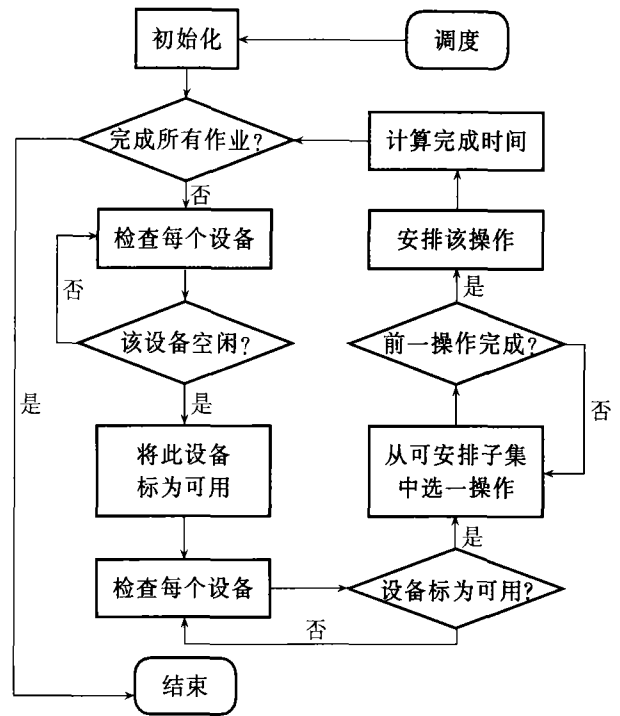


图 1 调度过程流程图

Fig.1 Flow diagram of scheduling process

表 1 不同机器台数和工件数的调度结果(时间参数)

样本编号	5 台机床									10 台机床								
	工件数/个									工件数/个								
	10			15			20			10			15			20		
遗传算法	启发式算法	改善率/%	遗传算法	启发式算法	改善率/%	遗传算法	启发式算法	改善率/%	遗传算法	启发式算法	改善率/%	遗传算法	启发式算法	改善率/%	遗传算法	启发式算法	改善率/%	
1	724	858	15.62	932	971	4.02	1 237	1 297	4.63	1 025	1 118	8.32	1 417	1 555	8.87	1 601	1 759	8.98
2	779	904	13.83	981	1 096	10.49	1 127	1 205	6.47	873	912	4.28	1 306	1 360	3.97	1 709	1 791	4.58
3	736	763	3.54	949	1 102	13.88	1 150	1 154	0.35	970	1 024	5.27	1 314	1 436	8.50	1 601	1 642	2.50
4	725	854	15.11	953	1 024	6.93	1 202	1 292	6.97	1 045	1 058	1.23	1 218	1 245	2.17	1 558	1 654	5.80
5	593	619	4.20	928	958	3.13	1 457	1 586	8.13	1 163	1 259	7.63	1 370	1 440	4.86	1 677	1 774	5.47

方法,不仅解决了工件排序时出现的死锁现象,同时也加快了遗传的收敛速度。仿真结果与 FCFS 调度方法所得出的结果进行了比较,比较结果证明了该算法的有效性。

参 考 文 献

[1] Davis L. Handbook of Genetic algorithm. New York: Van Nostrand Reinhold, 1991.
 [2] Xiong Rui, Fan Yushun, Wu Cheng. A dynamic job shop scheduling method based on lagrangian relaxation. Tsinghua Science and Technology, 1999, 4(1): 1 282-1 286.
 [3] Yamada T, Nakan R. A genetic algorithm applicable to

large-scale job-shop problems. Proc. of Parallel Problem Solving from Nature II, 1992: 281-290.

[4] 纪树新,钱积新,孙优贤. 车间作业调度遗传算法中的编码研究. 信息与控制,1997,26(5): 393-400.
 [5] 王万雷,李浙昆,樊瑜瑾.基于遗传算法的车间作业调度问题研究[D]. 云南:昆明理工大学,2002.
 [6] 陈荣秋. 生产计划与控制:概念、方法与系统.武汉:华中理工大学出版社,1995.
 [7] Leung L C, Maqneshwari S K, Miller W A. Concurrent part assignment and tool allocation in FMS with material handling considerations. Int. J. Prod. Res., 1993, 31(1): 117-138.

(责编 晓霖)