

并行机调度问题粒子群优化研究 *

刘志雄^{1 2 3}

(1 武汉科技大学 机械自动化学院 武汉 430081)

(2 天津港(集团)有限公司 博士后科研工作站 天津 300461)

(3 武汉理工大学 水路公路交通安全控制与装备教育部工程研究中心 武汉 430063)

Research of particle swarm optimization algorithm for parallel machine scheduling problem

LIU Zhi-xiong^{1 2 3}

(1 College of Machinery and Automation, Wuhan University of Science and Technology, Wuhan 430081, China)

(2 Tianjin Port (Group) Co., LTD Postdoctoral Research Center, Tianjin 300461, China)

(3 Engineering Research Center of Transportation Safety (Ministry of Education), Wuhan University of Technology, Wuhan 430063, China)

【摘 要】采用粒子群算法优化并行机调度问题,提出了基于机器和粒子位置取整的粒子编码方法和基于工件和粒子位置次序的粒子编码方法,并给出了两种不同粒子编码方法所对应的粒子群算法的步骤。通过对两个并行机算例的计算说明,基于两种不同编码方法的粒子群算法都能有效地对并行机调度问题进行优化,其中,基于工件和粒子位置次序的粒子编码所对应粒子群算法的优化性能要好些。

关键词 粒子群算法 并行机调度 粒子位置 编码

【Abstract】Particle swarm optimization algorithm is employed to optimize the parallel machine scheduling problem and two kinds of the encoding method are introduced including the encoding method based on machine and particle position rounding and the encoding method based on job and particle position sequence. Moreover the structures of particle swarm optimization algorithm based on two kinds of the encoding method are presented. From the computational results of two parallel machine scheduling problems particle swarm optimization algorithm based on two kinds of the encoding method can effectively optimize the parallel machine scheduling problems and the encoding method based on job and particle position sequence has better performance.

Key words Particle swarm optimization algorithm Parallel machine scheduling Particle position Encoding

中图分类号 :TH16 文献标识码 :A

1 引言

并行机调度(也称并行多机调度 Parallel Machines Scheduling Problem,简称 PMSP 问题)是实际生产过程中的一类典型调度问题,其研究 n 个工件 $\{J_1, \dots, J_n\}$ 在 m 台机器 $\{M_1, \dots, M_m\}$ 上的加工过程,每个工件只有一道工序^[1]。在并行机调度问题中,当每台机器均能满足各工件的加工时,此时的调度解主要考虑所有工件在各机器上的划分问题,或者说,如何将 m 台机器分配给 n 个工件。以最大完工时间最小为优化目标的并行机调度问题已经被证明是 NP 完全问题^[2-3]。

粒子群算法(Particle Swarm Optimization,简称 PSO)是一种基于群体智能的进化类算法^[4],一经提出便得到较为广泛的应用,特别是在连续函数优化问题领域,粒子群算法显示了其有效的优化性能^[5]。同时,通过对粒子群算法实施相应的编码和解码方法,以及对粒子群模型进行改进,粒子群算法也被应用于组合优化问题,如调度问题^[6-7]。

本文利用粒子群算法对并行机调度问题进行优化,针对并行机调度问题的特点,提出了基于机器和粒子位置取整的粒子编码方法以及基于工件和粒子位置次序的粒子编码方法,并对两种不同编码方法所对应的粒子群算法步骤进行了描述。最后采用基于两种不同编码方法的粒子群算法对两个不同规模的并行机调度问题算例进行了计算,计算结果说明两种不同粒子编码方法所对应的粒子群算法均能有效地对并行机调度问题进行优化。

2 并行机调度问题的描述

对于 n 个工件 m 台机器的并行机调度问题,如果不考虑同一台机器上加工各工件之间的准备时间,用 t_i 表示工件 i 的加工时间, W_i 表示加工过程中该工件等待加工的时间, C_i 表示该工件的加工完毕时间,用 f 表示所有工件的最大完工时间,则有

$$f = \max \{C_i \mid i=1, \dots, n\} = \min \max \{W_i + t_i \mid i=1, \dots, n\} \quad (1)$$

如果用 F 表示调度的优化目标,即最小化最大完工时间,则有 $F = \min f = \min \max \{C_i \mid i=1, \dots, n\} = \min \max \{W_i + t_i \mid i=1, \dots, n\}$ (2)

* 来稿日期:2010-01-21 * 基金项目:国家自然科学基金(70801047),中国博士后科研基金资助项目(20090450769)

并行机调度主要考虑的是所有工件在各机器上的划分(如果考虑工件的准备时间,调度问题的解还需要考虑各机器上工件的次序)。如果用集合 A 表示所有的工件集,即有 $A=\{J_1, \dots, J_n\}$, 用 A_i 表示 A 的一个子集, A_i 中的工件被分配给机器 M_i 进行加工, $i=1, 2, \dots, m$, $A=\bigcup_{i=1}^m A_i$, 并且 $\bigcap_{i=1}^m A_i=\phi$ (空集), 那么并行机调度的解就是要寻求集合 A 的 m 个子集 A_i 。

对于并行机调度问题,其可行解的前提条件是必须保证每台机器都被分配工件进行加工,在此基础上,对所有工件的任意分配都可行形成一个可行调度解。

3 粒子群算法设计

3.1 粒子群算法

在粒子群算法中,每个优化问题的解都用一个粒子表示,所有粒子都有一个由被优化的函数决定的适应值,每个粒子还有一个速度决定它们搜索的方向和距离,所有粒子通过追随当前的最优粒子(个体最优粒子和全局最优粒子)在解空间中进行搜索。假设粒子群中的第 i 个粒子在 n 维空间中的位置表示为一个 n 维向量 $x_i=(x_{i1}, x_{i2}, \dots, x_{in})$, 速度 $v_i=(v_{i1}, v_{i2}, \dots, v_{in})$ 决定了粒子在搜索空间单次迭代的位移。当前粒子群中的个体最优粒子表示为 $p_i=(p_{i1}, p_{i2}, \dots, p_{in})$, 全局最优粒子表示为 $g=(g_1, g_2, \dots, g_n)$, 粒子群中的所有粒子根据公式(3)和(4)来更新其速度和位置。

$$v_{ij}=\omega v_{ij}+c_1 \text{random}() (P_{ij}-x_{ij})+c_2 \text{random}() (g_j-x_{ij}) \quad (3)$$

$$x_{ij}=x_{ij}+v_{ij} \quad (4)$$

式中 $\text{random}()$ —(0,1)之间的随机数, c_1 和 c_2 —学习因子,一般取 $c_1=c_2=2$, ω —惯性权重, ω 取较大值粒子群算法具有较强的全局搜索能力, ω 取较小值粒子群算法倾向于局部搜索,一般的做法是将 ω 的初始值设置为 0.9, 然后按照迭代次数线性递减到 0.4。速度 v_i 可以设定在 v_{\max} 和 v_{\min} 之间, 当超过这个区间时, 就取对应的上下限值。 x_i 位置也可设定在一定的 $[x_{\min}, x_{\max}]$ 区间范围内。

3.2 粒子编码与解码方法

3.2.1 基于 Machine-PPR 的粒子编码

(1)粒子编码方法,定义一个二维粒子如下,二维粒子中的第一维用自然数 $1, 2, \dots, n$ 表示 n 个工件,第二维表示粒子的位置,粒子的长度为所有工件的数量。

对于粒子种群中第 i 个二维粒子 x_i 的一个向量可以表示为 $\begin{bmatrix} j \\ x_{ij} \end{bmatrix}$, $j=1, 2, \dots, n$, $ij \in [1, m+1]$, 是一个随机实数, m 为机器的数量,用自然数 $1, 2, \dots, m$ 表示不同的机器。整个二维粒子,如表 1 所示。

表 1 基于 Machine-PPR 的粒子编码

工件	1	2	3	...	n
位置	x_{i1}	x_{i2}	x_{i3}	...	x_{in}

在生成调度方案之前,需对上述二维粒子进行解码。对粒子的第二维即粒子位置 x_{ij} 进行取整操作,用 $\text{INT}(x_{ij})$ 表示,由于 $x_{ij} \in [1, m]$, 因此,取整后 $\text{INT}(x_{ij})$ 即表示工件 j 所对应的加工机

器。通过上述方式,将所有工件分配到不同的机器上,进而可以得到各机器上的工件加工序列,即调度方案。由于上述粒子编码和调度方案的生成方法是针对机器展开的,因此,本文将这种粒子表示称为基于机器和 PPR 的粒子编码。

(2)最大完工时间的计算,在并行机调度问题中,得到了上述各工件在各机器上的分配,就很容易计算所有工件的最大完工时间,如果不考虑工件的准备时间,那么,所有工件的最大完工时间的计算方法为:首先分别计算分配在各机器上的工件的加工时间的总和,然后取各机器上加工时间总和最大者即为所有工件的最大完工时间,即 Makespan。

(3)存在问题和修正方法,对于上述面向机器和 PPR 的粒子表示,在粒子的初始化阶段,粒子位置是 x_{ij} 在 $[1, m+1]$ 之间随机取值,然后通过对粒子位置 x_{ij} 的取整操作 $\text{INT}(x_{ij})$ 来表示该工件所对应的加工机器,从而生成调度方案,由于 $\text{INT}(x_{ij})$ 实质上是在 $[1, m]$ 之间取随机整数,因此就可能造成有些机器编号没有被分配给工件的结果,如表 2 所示。

表 2 粒子编码所映射的不可行调度方案

工件	1	2	3	4	5	6	7
机器	1	2	1	1	2	1	2

如表 2 所示,机器 3 没有被分配工件,这明显是一个不可行调度方案。另外,在粒子位置的更新阶段,粒子的位置被限制在 $[1, m+1]$ 之间,粒子的位置如果超出范围就分别取其范围的上下限值,这也可能造成上述不可行调度方案的生成。因此,本文采用如下方法对算法进行改进。

在粒子的初始化阶段,如果生成的粒子所对应的调度方案为不可行调度,即重新初始化该粒子,从而保证初始化粒子种群的可行性。

在粒子位置的更新阶段,采用一种变异算子,即如果粒子位置超出了其限制范围 $[1, m+1]$,不是将该粒子位置赋予上下限值,而是在 $[1, m+1]$ 之间随机取值。

采用一种惩罚的方法,即在计算过程中,如果某个粒子所生成的调度方案为不可行调度方案,将该粒子的适应值(即最大完工时间)设定为一个较大的值。

上述改进在一定程度上,至少是在粒子的初始化阶段,避免了不可行调度方案的生成,而惩罚方法则在进化过程中减少了对应着不可行调度方案的粒子被“选择”的机会,同时,变异算子的加入可以增加生成的调度方案的多样性。

3.2.2 PSO 算法步骤

综上所述,求解并行机调度的 PSO 算法的步骤如下: (1)设定粒子种群数量和最大迭代次数,对粒子群中的粒子 x_i 和 v_i 进行初始化。(2)对粒子进行解码并生成调度方案。计算粒子的适应值,即最大完工时间,如果某个粒子所生成的调度方案为不可行调度方案,将该粒子的适应值(即最大完工时间)设定为一个较大的值。(3)根据公式(3)和(4)对粒子群中粒子的位置和速度进行计算。如果被更新的粒子位置超过了限制范围 $[1, m+1]$,那么就在 $[1, m+1]$ 之间随机取值。(4)迭代是否结束,如果是,输出全局最优粒子和最小 Makespan,根据全局最优粒子生成最佳调度。

否则,转向步骤(2)。

3.3 基于 Job-PPS 的粒子编码

3.3.1 粒子编码

基于 Job-PPS 的粒子编码同样以二维粒子表示,第一维表示工件,第二维表示粒子位置。粒子编码形式与表 1 相同,与基于 Machine-PPR 的编码不同的是,表 1 中,粒子位置是面向机器数量取值的,即 $x_{ij} \in [1, m]$ 。而在基于 Job-PPS 的粒子编码中,粒子位置没有这种限制。整个粒子编码是面向工件的,即通过对粒子位置进行排序,从而间接对工件进行排序,然后,按照工件顺序依次将工件分配给每台机器,从而得到有效调度方案。

3.3.2 PSO 算法步骤

(1)设定粒子种群数量和最大迭代次数,对粒子群中的粒子 x_i 和 v_i 进行初始化。(2)对粒子进行解码,即对粒子位置进行排序,从而得到工件排序。(3)按照工件次序和机器的空闲状态,将工件进行分配到各机器上,同时计算该粒子的适应值,即调度方案的最大完工时间,具体实现过程如下。

令开始调度的时间,即时间 $T_s=0$ 。从调度方案即工件排序中,挑选排序最靠前的工件进行加工,并将其分配给目前空闲的机器,记录其加工结束时间,并将其放入加工队列。判断所有工件是否加工完毕,如果是,就结束调度,转向步骤,否则,说明有工件在等待加工(说明目前没有空闲的机器),就对加工队列中的加工结束时间进行排序,挑选出加工结束时间最早的工件,释放机器,并将时间 T_s 推进到该工件的加工结束时间,记录工件的等待时间,返回步骤。按照公式(1)计算最大完工时间,即对所有工件的加工结束时间进行排序,找出最大的加工结束时间,即为该调度方案所对应的最大完工时间。

(4)确定个体最优粒子和全局最优粒子。(5)根据公式(3)和(4)对粒子群中粒子的位置和速度进行计算。迭代是否结束,如果是,输出全局最优粒子和最小 Makespan,根据全局最优粒子生成最佳调度。否则,转向步骤②。

4 计算结果

为验证两种不同粒子编码方法及其粒子群算法的优化性能,这里采用文献[1]的两个不同规模的并行机调度问题算例,即 7 个工件 3 台机器和 30 个工件和 10 台机器两个算例,对基于上述两种不同粒子编码的粒子群算法进行测试。两个算例的加工数据,如表 3、表 4 所示。

表 3 7 个工件 3 台机器并行机问题的加工数据

工件	1	2	3	4	5	6	7
加工时间	6	6	4	4	4	3	3

表 4 30 个工件 10 台机器并行机问题的加工数据

工件	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
加工时间	3	2	6	4	5	7	9	13	4	12	10	18	22	11	8
工件	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
加工时间	26	14	6	17	27	11	17	26	16	7	23	15	18	15	13

采用惯性权重线性递减粒子群模型,粒子位置的初始化和限制范围为 $[1, m+1]$,粒子速度的初始化为 $[-2, 2]$ 。由于两个问题的规模区别较大,为了得到较好的计算结果。因此对于 7 工件

3 机器问题设定最大迭代次数为 100 次,对于 30 工件 10 机器问题设定最大迭代次数为 300 次,对于每个问题均连续优化 20 次,计算结果,如表 5 所示,其中 30 工件 10 台机器并行机问题 20 次优化的计算结果分布图,如图 1 所示。

表 5 两个不同并行机调度问题的 PSO 计算结果

算例	7 工件 3 机器问题		30 工件 10 机器问题	
不同编码	面向机器和 PPR 的编码	面向工件和 PPS 的编码	面向机器和 PPR 的编码	面向工件和 PPS 的编码
最优值	10	10	41	39
最差值	10	10	45	40
平均值	10	10	43.5	39.85

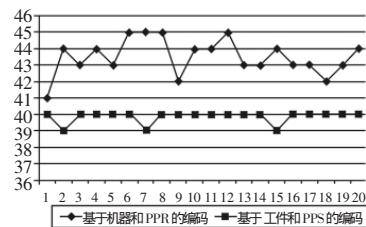


图 1 20 次优化的计算结果分布

对于 7 工件 3 机器的小规模问题,20 次连续优化运算中,基于两种不同粒子编码的粒子群算法均能找到最优值。

而对于 30 工件 10 机器的大规模问题,基于 Machine-PPR 粒子编码的 PSO 算法找到的最优 Makespan=41,最差 Makespan=45,平均值为 43.5,其最大偏差为 9.76%,平均偏差为 6.1%。基于 Job-PPS 粒子编码的 PSO 算法找到的最优 Makespan=39,平均值为 39.85,优化结果的最大偏差为 2.56%,平均偏差为 2.17%。因此,相比较而言,基于 Job-PPS 的粒子编码具有更好的优化性能。

5 结束语

针对并行机调度问题的粒子群算法进行了设计,首先提出了基于机器和 PPR 的粒子编码方法以及基于工件和 PPS 的粒子编码方法。针对在基于机器和 PPR 的粒子编码方法中容易产生不可行调度方案的缺点,对其粒子群算法进行了改进,改进后的算法在优化结果上有较好的提高。在此基础上,本文给出了基于两种不同粒子编码方法的粒子群算法步骤。

其次,将基于上述两种不同粒子表示方法的粒子群算法,对 7 工件 3 机器并行机调度问题和 30 工件 10 机器并行机调度问题两种不同规模的算例进行了测试,实验结果显示,基于两种不同粒子编码方法的粒子群算法均能够得到较好的优化结果,其中基于基于工件和 PPS 粒子编码的粒子群算法的优化性能较好,对于 30 工件 10 机器的并行机调度问题,其计算结果好于遗传算法的计算结果。

参考文献

- 唐恒永,赵传立.排序引论.北京:科学出版社,2002
- 郑大钟,赵千川.离散事件动态系统.北京:清华大学出版社,2001
- P. Brucker. Scheduling Algorithm (The Fifth Edition).Springer-Verlag, Berlin, 2007
- Kennedy J, Eberhart R C. Particle Swarm Optimization.Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, 1995, 1942~1948
- 雷秀娟,史忠科,付阿利.改进的粒子群优化算法求解车辆调度问题.计算机应用研究,2008,25(9):2674~2676
- 王凌.车间调度及其遗传算法.北京:清华大学出版社,2003