

基于改进粒子群算法的车间作业调度问题研究

乔佩利, 马丽丽, 郑 林

(哈尔滨理工大学 计算机科学与技术学院, 黑龙江 哈尔滨 150080)

摘 要: 针对现行的遗传算法存在过早收敛和进化速度过慢的局限, 以及标准粒子群算法收敛精确度不高、易陷入局部极值点的缺点, 通过分析原有算法的优化机理, 提出一种惯性权重随粒子的进化代数增加而非线性减小的改进型粒子群算法, 并将此算法应用于车间作业调度问题中. 大量仿真实验结果表明, 该算法在求解车间作业调度问题上具有可行性和有效性.

关键词: 粒子群算法; 车间作业调度问题; 惯性权重

中图分类号: TH 166 **文献标志码:** A **文章编号:** 1007-2683(2011)02-0035-05

Research on Job-shop Scheduling Problem Based on Improved Particle Swarm Optimization

QIAO Pei-li MA Li-li ZHENG Lin

(School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China)

Abstract Because the current genetic algorithms has the limitation of a premature convergence and the slow-evolutionary and the standard particle swarm optimization has the shortcomings of low convergence precision, through analyzing the mechanism of the original optimization algorithm, this article proposes an improved particle swarm optimization in which inertia weight non-linearly decreases with the increase of iterative generation, and applies this algorithm in job-shop scheduling problem. A large number of simulation results show that this algorithm has good feasibility and effectiveness in job-shop scheduling problem.

Key words particle swarm optimization; job-shop scheduling problem; inertia weight

0 引 言

车间作业调度问题 (job-shop scheduling problem, JSP) 是较旅行商问题更为复杂的典型调度问题, 是许多实际生产调度问题的简化模型, 具有很强的工程背景, 是一个典型的 NP-hard 问题, 是最困难的约束组合优化问题之一, 是生产计划与控制理论的重要组成部分^[1]. 目前, 它已经成为 CMS 领域中研究最广泛的一类经典调度问题, 因此有效调度方

法的研究具有重要的理论意义和工程价值.

车间作业调度问题是典型的组合优化问题, 求解该方法的方法有很多, 但主要以启发式算法为主. 近年来, 由于局部搜索方法的发展, 很多人开始研究使用局部搜索方法求解车间作业调度问题. 目前常用的方法有遗传算法、模拟退火算法、禁忌搜索算法以及这些方法的混合算法^[2]. 尽管这些研究取得了一定的应用效果, 但是还存在着许多问题, 如容易陷入局部最优、计算规模不可能较大等.

粒子群算法是近年来新出现的一种智能算法,

收稿日期: 2009-10-12

基金项目: 黑龙江省发展信息产业专项基金

作者简介: 乔佩利 (1951—), 男, 教授, 博士生导师, E-mail: qiaopeili@yahoo.com.cn

由于其具有操作简单、收敛速度快等优点,受到国内外众多学者的关注.标准的粒子群算法应用于 JSP 问题中,虽然也取得了一些成果,但存在容易陷入局部极值,容易早熟收敛等缺点.为此,国内外学者作了很多改进,包括对加速常数、进化公式、拓扑结构方面的改进,以及混合算法的研究等.针对该算法的不足,通过分析算法的优化机理,本文提出了一种惯性权重动态非线性变化的粒子群 (non-linear dynamic changing inertia weight particle swarm optimization, NDCW PSO) 算法,并将之应用于 JSP 问题中.

1 车间作业调度问题描述

车间作业调度就是根据已知各生产作业的加工时间及加工设备,在满足约束条件的限制下,使生产资源得到优化配置,以使生产作业的某项指标(所有机器的完工时间、工件流程时间、延迟时间及生产成本等)达到最优.车间作业调度一般可以描述为优化作业的加工顺序,使生产资源得到优化配置的过程.

JSP 问题可以简单描述为 n 个工件在 m 台机器上加工,每个工件包含 m 道工序,且已知对应每道工序在各自对应机器上的加工时间.对应的已知条件如下所述:

- 1) 加工的工件集 $O = (O_1, O_2, \dots, O_n)$, 其中 O_i 表示第 i 个工件, $i = 1, 2, \dots, n$;
- 2) 加工的机器集 $M = (m_1, m_2, \dots, m_m)$, 其中 m_i 表示第 i 个机器, $i = 1, 2, \dots, m$;
- 3) 各工件的工序集 $J = (J_1, J_2, \dots, J_n)^T$, 其中 $J_i = (J_{i1}, J_{i2}, \dots, J_{ik}, \dots, J_{im})$, 其中 J_{ik} 表示第 i 个工件的第 k 道工序, $i = 1, 2, \dots, n$, $k = 1, 2, \dots, m$;
- 4) 工件各工序的加工时间 $T = (T_1, T_2, \dots, T_n)^T$, $T_i = (T_{i1}, T_{i2}, \dots, T_{ik}, \dots, T_{im})$, 其中 T_{ik} 表示第 i 个工件的第 k 道工序的加工时间, $i = 1, 2, \dots, n$, $k = 1, 2, \dots, m$.

车间作业调度问题满足的约束条件如下:

- 1) 每个工件的加工工艺(即工件占用机器的先后顺序及加工时间)均是确定不变的;
- 2) 每个工件的下一道工序只能在上一道工序完成后才能开始加工;
- 3) 每台机器在同一时刻只能加工某一个工件的一道工序;
- 4) 每个工件都须经过 m 台机器加工;
- 5) 每台机器只要开始加工一道工序,本道工序

则不能中断,必须等到本道工序加工结束后,该机器才能加工其他工件;

6) 每个工件具有相同的加工优先级.

问题的目标函数是最小化最大完工时间,即

$$F = \min \{ \max_{1 \leq j \leq m} C_j \} \quad (1)$$

其中 C_j 表示第 j 号机器的最后完工时间.

2 粒子群优化算法

2.1 基本粒子群优化算法

粒子群优化算法 (particle swarm optimization PSO) 最早是由美国心理学研究者 Kennedy 博士和从事计算智能研究的 Eberhart 博士受鸟群觅食行为的启发于 1995 年提出的,并用于解决优化问题.算法采用速度-位置搜索模型.目前,PSO 算法已广泛应用于函数优化、神经网络训练、预测控制、模糊系统控制等应用领域中,并取得了良好的效果^[3].

Shi Y 和 Eberhart 于 1998 年对 Kennedy 的基本 PSO 模型进行了系统的改进,目前研究者大都采用了他们改进的模型^[4-5].假设在一个 d 维空间进行搜索,第 i 个微粒的位置就可以表示为一个 d 维向量, $X_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{id})^T$, 第 i 个微粒的速度 $V_i = (v_{i1}, v_{i2}, v_{i3}, \dots, v_{id})^T$, 第 i 个微粒经历过的最优位置(适应度最好的位置)记为 $P_{best} = (p_1, p_2, p_3, \dots, p_d)^T$, 群体所有微粒经历过的最优位置记为 $G_{best} = (g_1, g_2, g_3, \dots, g_d)^T$. 定义公式中的上标 T 表示迭代的次数,每一次迭代,微粒根据如下的公式来更新自己的速度和位置:

$$v_{ij}^{t+1} = \omega \times v_{ij}^t + c_1 \times \text{rand}() \times (p_j^t - x_{ij}^t) + c_2 \times \text{rand}() \times (g_j^t - x_{ij}^t) \quad (2)$$

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{T_{\max}} \times t \quad (3)$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \quad (4)$$

其中 $j = 1, 2, \dots, d$; $i = 1, 2, \dots, n$, n 为种群大小; t 为当前迭代次数; ω 为惯性权重,它使粒子保持运动惯性, $\omega_{\min}$ 、 $\omega_{\max}$ 分别为惯性权重的最小值和最大值, $T_{\max}$ 为最大迭代次数, ω 一般从 0.9 线性递减到 0.2; c_1 和 c_2 为两个正的加速度常数,称为学习因子,通常取 $c_1 = c_2 = 2 \times \text{rand}()$ 为 $[0, 1]$ 之间的随机数^[6-7].

2.2 改进的粒子群优化算法

标准 PSO 算法在解决车间作业调度问题时,虽

然有一定的效果,但仍然存在着收敛速度慢、容易陷入局部极值的缺点.在标准 PSO 算法中,惯性权重影响着算法的全局搜索和局部搜索能力.当 ω 较大时,微粒的全局搜索能力较强,当 ω 较小时,微粒的局部搜索能力较强.所以对惯性权重的改变,可以调节算法的全局和局部搜索能力^[8].目前常用的方法是采用线性递减 ω 策略,但是 PSO 算法的搜索过程是一个高度复杂的非线性过程,因此用线性策略得出的结果并不能反应实际的优化搜索过程.因此本文提出了一种惯性权重非线性动态变化的粒子群算法(简称 NDCW PSO),实验证明此算法比标准的 PSO 算法寻优能力强.通过引入动态非线性变化的 ω 后,粒子的位置更新公式不变,速度更新公式为

$$v_{ij}^{t+1} = \omega' \times v_{ij}^t + c_1 \times \text{rand}() \times (p_j^t - x_{ij}^t) + c_2 \times \text{rand}() \times (g_j^t - x_{ij}^t) \quad (5)$$

$$\omega = \omega_{\min} + (\omega_{\max} - \omega_{\min}) \exp\left(-\frac{t}{T_{\max} - t}\right) \quad (6)$$

其中: $\omega_{\min}$ 、 $\omega_{\max}$ 分别为惯性权重的最小值和最大值; $T_{\max}$ 为最大迭代次数; t 为当前迭代次数.大量仿真实验结果表明,当 $\omega_{\min} = 0.4$ $\omega_{\max} = 1.2$ 时,寻优能力最强.

3 用改进的粒子群算法求解 JSP 问题

3.1 粒子的编码

使用改进的粒子群优化算法 (NDCW PSO) 求解 JSP 问题的关键在于粒子的编码.本文针对 JSP 问题的特点,参照遗传算法解决 JSP 问题的方式,设计了一种基于工序的编码方式对粒子进行编码.这种编码方式把粒子向量中的每一个分量编码为工件号,向量中重复出现的同一工件号为同一工件的不同工序,根据工件号出现的顺序来区分是工件的第几个工序.例如,对于一个 3 台机器加工 3 个工件的问题,假设每个工件都有 3 个工序,随机产生一个操作顺序的编码, $X = [122313231]$, 第一个 1 代表工件 O_1 的第一个工序,接下来的 2 代表工件 O_2 的第一个工序,第三个位置的 2 代表工件 O_2 的第二个工序,依此类推,最后一个 1 代表工件 O_1 的第三个工序.这样,每个粒子代表工件的一个排序.但是,当粒子进行迭代更新时,改变的是粒子中各向量的值,而不是改变粒子中向量的排序.因此,需要对应每一个粒子设置一个向量 y_i , y_i 的各分量为粒子的位置权重,分别对应粒子中各工序的位置权重.通过 y_i 就可以间接地改变粒子中工件的排序了.

3.2 调度方案的生成

对粒子向量 Y_i 中的各分量从小到大排序,同时交换操作顺序 X 中对应的工序号.例如,以上述 3 台机器 3 个工件的 JSP 问题为例,假如某个粒子的向量如表 1 所示.

表 1 某粒子的向量

操作 X_i	位置 Y_i	操作 X_i	位置 Y_i
1	0.61	2	0.15
1	0.04	3	2.33
1	3.89	3	0.45
2	0.23	3	1.02
2	1.67		

对粒子中的 Y_i 向量按照从小到大的顺序进行排列,得到操作顺序如表 2 所示.

表 2 排列后操作顺序

位置 Y_i	操作 X_i	位置 Y_i	操作 X_i
0.04	1	1.02	3
0.15	2	1.67	2
0.23	2	2.33	3
0.45	3	3.89	1
0.61	1		

Y_i 在每一次迭代中发生位置的变化,对应生成不同的操作顺序 X_i ,这样就形成了不同的调度方案.

3.3 应用 NDCW PSO 算法求解 JSP 问题的步骤

根据调度方案的有序操作表,以所有机器的最大完工时间最小值为目标函数,具体求解步骤如下:

Step1 通过对问题的分析以及对解的特性的了解,设定粒子种群规模 N 和最大迭代次数 $T_{\max}$ 当前迭代次数 t 的大小.

Step2 用随机函数法初始化粒子群,随机产生粒子的位置 X_i 和速度 V_i .

Step3 对粒子位置向量各分量进行排序,利用本文提出的编码方式对每个粒子进行编码,结合工件加工的约束条件生成有序操作表.

Step4 对有序操作表进行解码,生成调度方案,计算粒子的适应度值,记录粒子的全局最优值 G_{best} 和个体最优值 P_{best} (G_{best} 初始化为所有粒子第一次迭代时产生的适应值中的最好适应值, P_{best} 初始化为每个粒子第一次迭代的适应值),具体的解码过程如下:

对于一个加工工艺表已知的 JSP 问题,给每个

粒子分别设置两个向量 T_{start} 和 T_{end} , T_{start} 用来记录各个工序的加工开始时刻, T_{end} 用来记录各个工序的加工结束时刻, 另外设置向量 T_{add} 用来记录各个机器的当前累计加工时间 (T_{add} 初始化为零向量).

1) 从粒子向量中顺序读取一个工件号, 查找加工工艺表, 确定该工序所需要的加工时间和所使用的机器, 将该机器的加工累计时间和该机器上前一工序的加工结束时间进行比较, 取其中较大者作为该工序的加工开始时刻, 并记录到 T_{start} 中, 再次查找加工工艺表, 计算出该工序的加工结束时刻, 并记录到 T_{end} 中, 同时更新该机器的加工累计时间 T_{add} .

2) 顺序读取下一个工序号, 继续执行 1) 中的操作.

3) 当粒子中最后一个工序号处理结束时, T_{add} 分量中的最大值即为粒子的适应度值.

4) 计算所有粒子的适应度值, 最大完工时间最小的那个粒子为当前最优粒子, 将当前最优适应值与 G_{best} 比较, 如果比 G_{best} 小, 则将当前最优适应值设置为 G_{best} , 否则, G_{best} 不变.

5) 对于每个粒子, 将其当前适应值与自身 P_{best} 比较, 如果比 P_{best} 小, 则将当前最优适应值设置为 P_{best} , 否则, P_{best} 不变.

Step5 由公式 (5)、(6) 和式 (4) 对粒子的速度和位置进行更新.

Step6 判断 t 是否达到最大迭代次数 T_{max} . 如果未达到, 则 $t = t + 1$ 转 Step3 否则, 转 Step7

Step7 输出全局最优值 G_{best} 所对应的粒子的工序序列, 即为最优排序结果.

3.4 算法复杂性分析

算法复杂性主要源于粒子群算法的基本步骤, 假设粒子的维数为 D , 数目为 N , 则粒子初始化的时间复杂度为 $O(ND)$, 速度位置更新的复杂度为 $O(N)$, 判断终止条件的复杂度为常数, 因此最大值为 $O(ND)$, 假设迭代次数为 T , 则执行完上面的循环后, 算法的复杂度为 $O(TND)$, 因此粒子群算法的时间复杂度为 $O(TND)$.

4 仿真结果及分析

为了验证改进算法的有效性, 本文首先以 FT06 问题为测试实例, 以 VC++ 6.0 为开发环境, 分别使用标准 PSO 算法和改进的 NDCW PSO 算法求解该问题. FT06 问题是 6 个工件在 6 台机器上加工的问题, 每个工件有 6 个工序, 工序总数为 36, 其加工工

艺表如表 3 所示.

表 3 FT06 的加工工艺表

工件	机器编号 加工时间 (s)					
	工序 6	工序 1	工序 2	工序 3	工序 4	工序 5
1	3/1	1/3	2/6	4/7	6/3	5/6
2	2/8	3/5	5/10	6/10	1/10	4/4
3	3/5	4/4	6/8	1/9	2/1	5/7
4	2/5	1/5	3/5	4/3	5/8	6/9
5	3/9	2/3	5/5	6/4	1/3	4/1
6	2/3	4/3	6/9	1/10	5/4	3/1

算法参数设置如下: 粒子个数 $N = 30$ 最大迭代次数 $T_{max} = 600$ 当前迭代次数 $t = 0$ 学习因子 $c_1 = c_2 = 2$ NDCW PSO 算法中惯性权重最小值 $\omega_{min} = 0.4$ 惯性权重最大值 $\omega_{max} = 1.2$ 标准 PSO 算法中惯性权重最小值 $\omega_{min} = 0.2$ 惯性权重最大值 $\omega_{max} = 0.9$ 每种算法分别运行十次, 求得的最优解、CPU 运行时间、收敛代数、平均值、收敛率如表 4 所示, 标准 PSO 算法的工序调度最优结果甘特图如图 1 所示, 改进算法的工序调度最优结果甘特图如图 2 所示.

表 4 标准 PSO 和 NDCW PSO 算法结果比较

次数	标准 PSO 算法			NDCW PSO 算法		
	最优解	CPU 运行时间 /ms	收敛代数	最优解	CPU 运行时间 /ms	收敛代数
1	56	3 298	458	56	602.4	234
2	57	3 450	441	55	5 783	242
3	57	3 302	462	55	5 628	256
4	55	3 451	459	55	5 807	261
5	55	3 507	463	55	5 241	225
6	58	3 287	471	57	5 409	272
7	56	3 514	447	55	5 667	268
8	59	3 316	457	56	5 754	259
9	55	3 382	450	55	5 904	247
10	57	3 464	468	55	5 260	238
平均值	56.5	3 397.1	457.6	55.4	5 647.7	250.2
收敛率	3/10			7/10		

为了验证改进算法的通用性, 本文又选取 LA 类问题为测试实例, 实验参数设置同上, 对 LA1 问题进行求解得到的结果统计如表 5 所示.

表 5 标准 PSO 和 NDCW PSO 算法求解 LA1 问题的结果比较

次数	标准 PSO 算法			NDCW PSO 算法		
	最优解	偏差 $\%$	收敛代数	最优解	偏差 $\%$	收敛代数
1	689	3.45	465	669	0.45	252
2	688	3.30	477	666	0	256
3	666	0	489	666	0	270
4	697	4.65	483	666	0	267
5	699	4.95	490	666	0	245
6	670	0.60	467	666	0	253
7	666	0	451	666	0	261
8	673	1.05	458	679	1.95	239
9	668	0.30	462	666	0	248
10	688	3.30	476	666	0	266
平均值	680.4	2.16	471.8	667.6	0.24	255.5

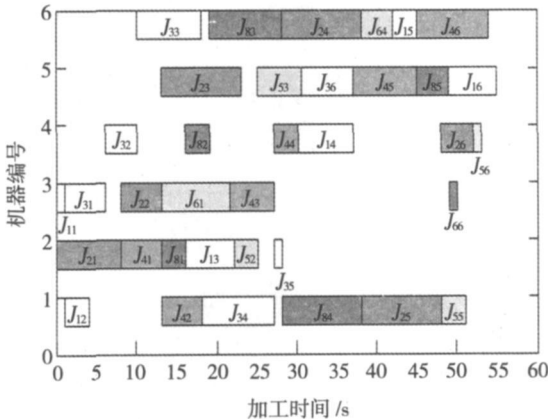


图 1 标准 PSO 算法的工序调度最优结果甘特图

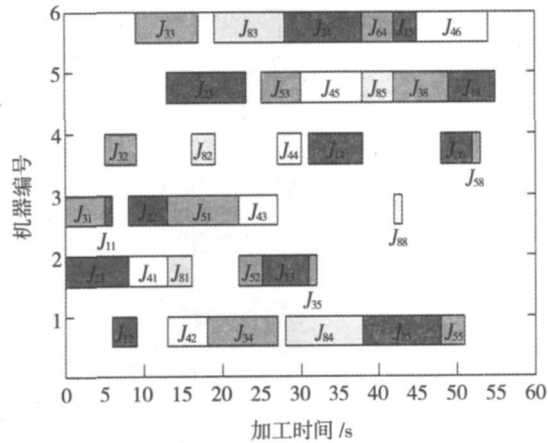


图 2 改进算法的工序调度最优结果甘特图

从表 4 和表 5 的算法结果比较, 可以看出

NDCW PSO 算法找到的最优解好于标准 PSO 算法的, 而且收敛速度也要优于标准 PSO 算法, 由于引入了动态惯性权重策略, 速度位置更新较之前的算法复杂, 导致计算时间长了点, 但是相对于其他方面的改进, 这点代价还是值得的。

图 1 和图 2 中横坐标代表加工时间, 纵坐标代表机器编号, J_{ik} 代表第 i 个工件的第 k 道工序, 虽然两种算法都可以找到最优解 55, 但是从表 4 和表 5 可以明显的看出, 改进 PSO 算法的效率明显高于标准 PSO 算法, 证明了改进算法的有效性。

5 总 结

本文在标准 PSO 算法的基础上, 针对车间作业调度问题的特点, 提出了一种惯性权重动态非线性变化的粒子群算法, 通过对 FT06 问题的求解, 验证了该算法求解车间作业调度问题的可行性与有效性。

参 考 文 献:

[1] 王凌. 车间调度及其遗传算法 [M]. 北京: 清华大学出版社, 2003 71-121

[2] 梁旭, 董明, 常征. 求解车间调度问题的一种新遗传退火混合策略 [J]. 计算机集成制造系统, 2005, 11(6): 851-854

[3] 曾建潮, 介婧, 崔志华. 微粒群算法 [M]. 北京: 科学出版社, 2004: 154-198

[4] 宋胜利, 孔力, 程晶晶. 混合优化算法在氧化铝生产中物料平衡计算上的应用 [J]. 仪表技术与传感器. 2009 9(9): 106-108

[5] CLERC M. Discrete Particle Swarm Optimization [J]. New Optimization Techniques in Engineering Springer-Verlag 2004, 23(2): 15-24

[6] GRAVES S A Review of Production Scheduling [J]. Operations Research, 2008 29(4): 646-675

[7] EBERHART R, SHI Y. Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization [C] // IEEE Congress on Evolutionary Computation. Piscataway: IEEE Service Center 2000: 84-88

[8] CHATTERJEE A, SIARRY P. Nonlinear Inertia Weight Variation for Dynamic Adaptation in Particle Swarm Optimization [J]. Computers & Operations Research, 2006(33): 859-871.

(编辑: 付长纓)