文章编号:1002-0411(2005)03-0365-04

基于自适应变异的粒子群优化算法的车间作业调度优化及其软件实现

那 加
(东南大学经济管理学院,江苏 南京 210096)

摘 要:由于现行的遗传算法在解决车间作业调度问题时有局限性,本文将一个自适应变异的粒子优化算法应用于车间作业调度.该算法在运行的过程中根据群体适应度方差以及当前最优解的大小来确定当前最佳粒子的变异概率,变异操作增强了粒子群优化算法跳出局部最优解的能力.仿真实例的结果表明:该算法在解决车间作业调度问题上是可行的.

关键词:粒子群;自适应变异;车间作业调度
中图分类号:TP273 文献标识码:A

Application of Particle Swarm Optimization with Adaptive Mutation to Job Shop Scheduling Problem and Its Software Implementation

NA Jia
(*School of Economics and Management, Southeast University, Nanjing* 210096, *China*)

Abstract: The reason why genetic algorithm available exhibits limitations when it is applied to job-shop scheduling problem (JSSP) is analyzed. In this paper, a new particle swarm optimization algorithm is applied to solve the problems in the JSSP. During the running, the mutation probability for the current best particle is determined by two factors: the variance of the population's fitness and the current optimal solution. The ability of particle swarm optimization algorithm(PSO) to break away from the local optimum is greatly improved by the mutation. The results of the example verify its better performance compared with the conventional algorithms.

Keywords: particle swarm; adaptive mutation; job-shop scheduling

1 引言 (Introduction)

车间作业调度问题(Job-Shop Scheduling Problem, JSSP)是生产管理中的一个重要的研究课题.它是一项加工资源分配问题,它根据约束条件,合理安排资源、加工时间、加工顺序以获得最优的成本或效率.在众多解决这一问题的方法中,以遗传算法为代表的进化算法在该领域获得了广泛应用.但遗传算法在解决大规模的JSSP时存在两大局限[1~3],即进化速度过慢和过早收敛(早熟).针对这一问题,人们应用粒子群优化(Particle Swarm Optimization, PSO)算法,这一算法在一定程度上克服了遗传算法的两个缺陷,但不能从根本上解决早熟收敛,同时使计算的工作量加大了许多.在这篇文章中,将应用一种新的基于群体适应度方差自适应变异的粒子群优化算法[5](Adaptive Mutation Particle Swarm Optimization, AMPSO)来解决ISSP,且用MATLAB编程来实现.

2 JSSP的数学模型 (Model of JSSP)

2.1 问题描述

JSSP问题可描述为:m台机器(用集合$M=\{M_j\}_{j=1}^{m}$表示)加工n个工件(用集合$J=\{J_i\}_{i=1}^{n}$表示),每个工件包含由多道工序组成的一个工序集合.工件有预先确定的加工顺序,每道工序的加工时间t_{ij},在给定的时间内每个机器只能加工一个工件,并且每个工件只能由一台机器处理.不同工件的加工顺序无限制,工序不允许中断;要求在可行调度中确定每个工序的开始时间s_{ij},使总完工时间$C_{\max}$最小,即$C^*_{\max}=\min(C_{\max})=\min\{\max(s_{ij}+t_{ij});\forall J_i$

收稿日期:2005-01-25

$\in J, M_j \in M\}$,求解满足以上条件的工件加工顺序即构成 JSSP调度问题.

2.2 整数规划 (IP)模型

整数规划模型由 Baker提出[1],需要考虑两类约束:工件工序的前后约束和工序的非堵塞约束.用 t_{jk}和 c_{jk}分别表示工件 j在机器 k上的加工时间和完工时间.如果机器 h上的工件加工工序先于机器 k (用 $J_h \prec J_k$表示),则有关系式 $c_{jk} - t_{jk} \geqslant c_{jh}$;反之,如果 $J_k \prec J_h$,有 $c_{jh} - t_{jh} \geqslant c_{jk}$.定义指示系数,$a_{ihk} = \begin{cases} 1, & J_h \prec J_k \\ 0, & \text{其它} \end{cases}$, $x_{ijk} = \begin{cases} 1, & i\text{先于 } j\text{到达机器 } k \\ 0, & \text{其它} \end{cases}$, M为一个大数,则工序的前后约束表示为:$c_{ik} - t_{ik} + M(1 - a_{ihk}) \geqslant c_{ih}$;工序的非堵塞约束表示为:$c_{jk} - c_{ik} + M \cdot (1 - x_{ijk}) \geqslant t_{jk}$, $i, j = 1, 2, \cdots, n; k = 1, 2, \cdots, m$,以 $C_{\max}$为目标的 IP模型可以表示为:

$$\min \max_{1 \leqslant k \leqslant m} \{c_{ik}\} \tag{1}$$

$$\text{s.t. } c_{ik} - t_{ik} + M(1 - a_{ihk}) \geqslant c_{ih}$$

$$c_{jk} - c_{ik} + M(1 - x_{ijk}) \geqslant t_{jk}$$

如果以平均流程时间为目标函数,可以改为 $\min \frac{1}{n} \sum_{i=1}^{n} \max_{1 \leqslant k \leqslant m} \{c_{ik}\}$,大数 M在可行区域范围内的取值由学者 Van Hulle给出:$M > \{\sum_{i=1}^{n} \sum_{k=1}^{m} t_{ik} - \min(t_{ik})\}$.

3 自适应变异的粒子群优化算法的描述 (Description of particle swarm optimization algorithm with adaptive mutation)

在粒子群优化算法的运行过程中,如果群体适应度方差等于零,且此时得到的最优解不是理论最优解或者期望最优解 f_d,则粒子群陷入局部最优,算法将出现早熟收敛.自适应变异的粒子群优化算法基于群体适应方差,对全局极值 g_{Best}进行自适应变异,使算法在发生早熟收敛时,能够跳出局部最优,进入解空间的其它区域继续进行搜索,直到最后找到全部最优解[5].

算法的步骤:

Step 1:随机初始化粒子的位置与速度.

Step 2:将粒子的个体极限 p_{Best}设置为当前位置,全局极值 g_{Best}设置为初始群体中最佳粒子位置.

Step 3:判断算法收敛准则是否满足,如果满足,转向 Step 9;否则,执行 Step 4.

Step 4:对粒子群中的所有粒子,执行如下操作:

1) 根据下列式 (1)、(2)、(3)更新粒子的位置与速度.

$$V = wV + c_1 \cdot rnd \cdot (p_{\text{Best}} - Present) + c \cdot rnd \cdot (g_{\text{Best}} - Present) \tag{2}$$

$$Present = Present + V \tag{3}$$

$$w = w_{\max} - run \frac{(w_{\max} - w_{\min})}{runmax} \tag{4}$$

其中:V是粒子的速度;$Present$是粒子的当前位置;rnd是 [0,1]之间的随机数;c_1 和 c_2 被称作学习因子.通常,$c_1 = c_2 = 2$;w是加权系数,一般在 0.1到0.9之间取值,通过大量实验证明,如果 w随算法迭代的进行而线性减少,将显著改善算法的收敛性能,设 $w_{\max}$、$w_{\min}$分别为最大、最小加权系数,run为当前迭代次数,$runMax$为算法迭代总次数.

2) 如果粒子适应度优于个体极值 p_{Best}的适应度,p_{Best}设置为新位置.

3) 如果粒子速度优于 g_{Best}的适应度,g_{Best}设置为新位置.

Step 5:根据下列式 (4)、(5)计算群体适应度方差 σ^2,并计算 $f(g_{\text{Best}})$.

$$\sigma^2 = \sum_{i=1}^{n} \left(\frac{f_i - f_{\text{avg}}}{f}\right)^2 \tag{5}$$

$$f = \begin{cases} \max\{|f_i - f_{\text{avg}}|\}, & \max\{|f_i - f_{\text{avg}}|\} > 1 \\ 1, & \text{其它} \end{cases} \tag{6}$$

其中,粒子群的粒子数目为 n, f_i为第 i个粒子的适应度, f_{avg}为粒子群目前的平均适应度; f是归一化定标因子,其作用是限制 σ^2 的大小,f可以取任意值,但要注意两个条件:① 归一化后,整个粒子群 $|f_i - f_{\text{avg}}|$的最大值不大于 1;② f随算法的进化而变化.

σ^2 反映的是粒子群中所有粒子的“收敛”程度,σ^2 越小,则粒子群越趋于收敛;反之,粒子群则处于随机搜索阶段.

Step 6:根据式 (6)计算变异概率 P_m

$$P_m = \begin{cases} k, & \sigma^2 < \sigma_d^2 \text{且 } f(g_{\text{Best}}) > f_d \\ 0, & \text{其它} \end{cases} \tag{7}$$

其中,k可取 [0.1, 0.3]之间的任意数值.σ_d^2 的取值与实际问题有关,一般远小于 σ^2 的最大值.f_d可以设置为理论最优值.

Step 7:产生随机数 $r \in [0,1]$,如果 $r < P_m$,按式 (7)执行变异操作,否则,转向 Step 8.

$$g_{\text{Best}_k} = g_{\text{Best}_k}(1 + 0.5\eta) \tag{8}$$

其中,g_{Best_k}为 g_{Best}的第 k维取值,η是服从

Gauss(0,1)分布的随机变量.

Step 8:判断算法收敛准则是否满足,如满足,执行 Step 9;否则,转向 Step 4.

Step 9:输出,算法运行结束.

4　算例仿真 (Simulation example)

4.1　编码

按所有工件的所有加工工序进行编码,即对于工序 $i(i=1,2,\cdots,n,n$表示工件的总数$)$,首先明确它的工序序数 $l_j(j=1,2,\cdots,m,m$为机器总数$)$,并用自然数表示;在染色体形成中,一个工件的出现次序表示该工件的第 n道加工工序.对于 m台机器 n个工件问题,一个染色体包括 $n\times m$个基因.

4.2　解码

对经过算法运算后生成的染色体码串,从第一个基因开始,依次排列到相应的机器上,即可得到一个有效解.

4.3　用 MATLAB语言编程

下面给出 AMPSO算法流程:

```
for    iter=1:runMax
  for popIndex=1:popsize
    评价各粒子的适应度
    if粒子适应度 > objectFun(pBest)
      pBest =粒子当前位置
    end
    if粒子适应度 > objectFun(gBest)
      gBest =粒子当前位置
    end
  粒子速度更新,粒子位置更新,计算适应度方差 σ
  计算粒子适应值 f(gBest);并计算变异概率 Pm
  end
  if  随机数 r<变异概率
    gBest_k = gBest_k (1 +0.5η)
  end
  if  满足收敛条件
  end
  输出 gBest
end
```

根据文中第 3节描述的自适应变异的粒子群优化算法的步骤,用 MATLAB语言编程,生成 AMPSO.m文件.根据表 1中的原始数据,运行 AMPSO.m文件,得出的结果如表 2、3.

表 1　车间作业调度仿真实例的原始数据

Tab.1　Original simulation data of JSSP

工件	工序	机器 1	机器 2	机器 3	机器 4	机器 5	机器 6
工件 1	O_{11}	2	3	4			
	O_{12}		3		2	4	
	O_{13}	1	4	5			
工件 2	O_{21}	3		5		2	
	O_{22}	4	3			6	
	O_{23}			4		7	11
工件 3	O_{31}	5	6				
	O_{32}		4		3	5	
	O_{33}			13		9	12
工件 4	O_{41}	9		7	9		
	O_{42}		6		4		5
	O_{43}	1		3			3

表 2　工件—工序在机器上的排序

Tab.2　Work pieces-working procedures compositor on the machines

次序 / 机器	1	2	3	4	5	6	7
机器 1	(1,1)	(1,3)	(2,1)	(2,2)	(3,1)	(4,1)	(4,3)
机器 2	(1,1)	(1,2)	(1,3)	(2,2)	(3,1)	(3,2)	(4,2)
机器 3	(1,1)	(1,3)	(2,1)	(2,3)	(3,3)	(4,1)	(4,3)

机器 4	(1,2)	(3,2)	(4,1)	(4,2)			
机器 5	(1,2)	(2,1)	(2,2)	(2,3)	(3,2)	(3,3)	
机器 6	(2,3)	(3,3)	(4,2)	(4,3)			

注:表 2中 (* ,*)表示工件及其对应的工序

表 3 工件—工序的加工开始和结束时间

Tab. 3 Start and finish machining time of work pieces-working procedures

机器 \ 次序	1	2	3	4	5	6	7
机器 1	(19,21)	(21,22)	(23,26)	(31,35)	(26,31)	(0,9)	(9,10)
机器 2	(9,21)	(12,15)	(15,19)	(26,29)	(31,37)	(37,41)	(19,25)
机器 3	(0,4)	(4,9)	(35,40)	(40,44)	(44,57)	(25,32)	(32,35)
机器 4	(19,21)	(41,44)	(10,19)	(35,39)			
机器 5	(21,25)	(14,16)	(16,23)	(23,30)	(0,5)	(5,14)	
机器 6	(0,11)	(14,26)	(39,44)	(44,47)			

注:表 3中 (* ,*)表示表 2中的工件对应工序的加工开始和结束时间.

应用这一算法可以产生的近似最优解较多,由于受篇幅限制,仅选一个近似最优解.近似最优解的时间周期为 57min.

5 结束语 (Conculsion)

自适应变异的粒子群优化算法与遗传算法相比没有选择、交叉等操作,与粒子群优化算法相比多了个变异操作,所以算法结构比遗传算法简单,且运行速度快,比遗传算法快 10~12%左右,比粒子群优化算法运行速度略为慢 3~4%,但自适应变异的粒子群优化算法不会出现所谓的早收敛(早熟)现象.该算法满足车间作业调度的在线生产要求,且取得较好的优化效果.

参 考 文 献 (References)

[1] 王书锋,邹益仁.车间作业调度(JSSP)技术问题简明综述[J].系统工程理论与实践,2003,27(1):49~54.

[2] 王书振,等.嫁接遗传算法及其在车间作业调度中的应用[J].机械科学与技术,2003,22(6):873~876.

[3] 周 弘,姬 彬.求解作业排序问题的通用混合遗传算法研究[J]. 系统工程理论与实践,2001, 25(12):66~71.

[4] 赵 虎,李 瞿.蚂蚁算法在车间作业调度问题中的应用[J]. 计算机工程与应用,2003,39(22):6~8.

[5] 吕振肃,侯志荣.自适应变异的粒子群优化算法[J].电子学报,2004,32(3):415~419.

[6] 侯志荣,吕振肃. 基于 MATLAB的粒子群优化算法及其应用[J]. 计算机仿真,2003,20(10):68~70.

[7] van den Bergh F, Engelbrecht A P. Cooperative learning in neural networks using particle swarm optimizers[J]. South African Computer Journal,2000,(26):84~90.

[8] 王万良,吴启迪,宋毅.求解作业车间调度问题的改进自适应遗传算法[J].系统工程理论与实践,2004,28(2):58~62.

作者简介

那 加(1958-),男,博士研究生,高级工程师.研究领域为生产运作管理.