

# 带公共交货期的并行机提前/拖后调度问题的进化策略求解

杨英杰, 刘 民, 吴 澄

(清华大学自动化系, 北京 100084)

**摘 要:** 本文首次将进化策略应用于带公共交货期的并行机提前/拖后调度问题, 在问题描述、个体的复合编码表示、进化策略参数的模糊逻辑动态控制和进化策略的局部寻优方面作了研究. 不同规模的计算实例表明本文提出的进化策略算法可以有效解决较大规模并行机调度问题.

**关键词:** 并行机; 调度; 进化策略; 复合编码; 模糊逻辑

**中图分类号:** TP18; O221 **文献标识码:** A **文章编号:** 0372-2112(2001)11-1478-04

## Evolutionary Strategy Method for Parallel Machine Earliness/Tardiness Scheduling Problem with Common Due Date

YANG Ying-jie, LIU Min, WU Cheng

(Department of Automation, Tsinghua University, Beijing 100084, China)

**Abstract:** The Evolutionary strategy method is first introduced into parallel machine earliness/tardiness scheduling problem with common due date. Researches are made in the aspects of problem formulation, composite code of individual, dynamic fuzzy logic control of parameters of evolutionary strategy and local searching. Computational results of different scale problems show that the method can effectively solve the parallel machine scheduling problem with relatively large scale.

**Key words:** parallel machine; scheduling; evolutionary strategy; composite code; fuzzy logic

### 1 引言

随着准时生产(JIT)模式在日本制造业中获得成功, 近几年来以准时生产为目标的提前/拖后调度问题已成为一个活跃的研究领域. 在带公共交货期约束的提前/拖后调度问题中, 若公共交货期待定, 则需要同时确定最优公共交货期和最优调度方案, 对这类问题的文献报道不多, 且主要是一些启发式算法<sup>[2~4]</sup>, 但启发式算法随着问题规模的扩大, 求得的解的质量不令人满意. 进化策略思想源于生物进化的突变和自然选择, 其主要特征是高斯变异和确定性选择, 由 Rechenbe 在 1964 年研究流体动力学问题时提出<sup>[1]</sup>. 直到 90 年代, 学术界才开始重视进化策略, 但与遗传算法<sup>[5~8]</sup>和启发式算法相比, 进化策略的应用还不多, 特别是在生产调度领域. 本文第一次将进化策略应用于有待定公共交货期的并行机提前/拖后调度问题, 在问题描述、解的表示、进化策略参数的动态模糊控制和进化策略的局部寻优方面作了研究, 并用实例进行了验证. 不同规模的实例表明用进化策略求解是有效的, 且解的质量优于启发式算法.

### 2 问题描述

在 JIT 生产模式下, 按用户的要求准时地完成加工任务是极为重要的. 对具有待定公共交货期的并行机器生产线提前/拖后调度问题可描述如下: 生产线由  $m$  台完全相同的机器组成, 有  $n$  个相互独立的任务要在生产线上加工, 每个任务都有确定的加工时间和待定的公共交货期, 均可由台机器中的任一完成加工. 要找一个最优公共交货期和最优调度方案, 即确定公共交货期和每台机器上加工的任务代号及其加工顺序, 使加工完所有任务后交货期安排成本、提前交货成本和拖期交货成本的总和最小.

设机器数和任务数分别为  $m$  和  $n$ ,  $T(i)$  为任务  $i$  的加工时间,  $C(i)$  为任务  $i$  的完工时间,  $d$  为任务的公共交货期, 则对任务  $i$ , 拖后时间为  $Tardy(i) = \max[C(i) - d, 0]$ , 提前时间为  $Early(i) = \max[d - C(i), 0]$ .

调度目标是分配  $n$  个任务给  $m$  台机器, 使加工完所有任务后由下式给出的惩罚函数  $f$  为最小:

$$f = \sum_{i=1}^n [\alpha d + \beta \times Early(i) + \gamma \times Tardy(i)]$$

$$= \alpha d + \sum_{i=1}^n [\beta \times \text{Early}(i) + \gamma \times \text{Tardy}(i)] \quad (1)$$

显然  $f$  是调度方案  $\delta$  和公共交货期  $d$  的函数。在式(1)中,  $\alpha, \beta, \gamma$  为非负常数, 分别表示交货期安排、提前交货和拖后交货的单位成本。

### 3 进化策略算法设计

#### (1) 编码和译码

进化策略的个体是多维目标向量, 在本文中, 目标向量分成四段。为提高进化策略的性能和效率, 对第一段向量提出实数编码与整数编码相结合的复合编码方式, 前者用于算子操作, 后者用于译码; 后三段向量分别采用单一的编码形式, 即由  $n$  个  $(0, 1)$  之间的实数组成第一段向量  $x_1, x_2, \dots, x_j, \dots, x_n$ , 用于算子操作; 第二段向量为公共交货期; 第三段向量为第一段向量相应的变异策略参数; 第四段向量为第二段向量相应的变异策略参数。在进行算子操作时, 个体的编码为  $x_1, x_2, \dots, x_j, \dots, x_n | d | \sigma_1, \sigma_2, \dots, \sigma_j, \dots, \sigma_n | \sigma_d$ , 总位数为  $2n+2$ 。

用于译码的个体第一段向量的编码按以下方式产生: 将上述第一段向量  $x_1, x_2, \dots, x_j, \dots, x_n$  进行递减排序, 再将  $x_j$  的序号  $k_j$  组成整数序列  $k_1, k_2, \dots, k_j, \dots, k_n$  (其中  $k_j, j \in [1, n]$  为互不相同的自然数)。该编码作为并行机的加工任务序列用于译码过程。具体译码过程为: 每次调度时, 根据  $k_1, k_2, \dots, k_j, \dots, k_n$  由左至右的顺序依次安排任务进行加工; 有多台空闲机器时, 按机器编号  $1 \rightarrow 2 \rightarrow \dots \rightarrow m$  的顺序选择相应的机器安排加工任务。通过以上译码可得到调度方案, 并求出相应的目标函数。

本文提出的复合编码方式, 既方便译码, 又有利于提高算子的操作效率。

#### (2) 产生初始种群

随机产生由  $\mu$  个个体组成的初始种群  $P(0) = \{a_1(0), a_2(0), \dots, a_\mu(0)\}$ , 其中  $a_l(k) = x_1, x_2, \dots, x_j, \dots, x_n | d | \sigma_1, \sigma_2, \dots, \sigma_j, \dots, \sigma_n | \sigma_d$ ,  $1 \leq l \leq \mu$ ,  $k \geq 0$  表示第  $k$  代的第  $l$  个个体。对公共交货期  $d$ , 取值为  $(0, d_{\max})$  间的随机数, 其中  $d_{\max} = \sum_{j=1}^n T(k_j) / n + \max_{1 \leq l \leq n} T(k_l)$ 。通过限定  $d$  的范围, 缩小了个体的初始可行解空间, 有利产生较优的初始可行解, 提高算法效率。

#### (3) 变异

进化策略中, 变异算子采用高斯扰动。个体  $x_1, x_2, \dots, x_j, \dots, x_n | d | \sigma_1, \sigma_2, \dots, \sigma_j, \dots, \sigma_n | \sigma_d$  在变异算子作用下变为  $x'_1, x'_2, \dots, x'_j, \dots, x'_n | d' | \sigma'_1, \sigma'_2, \dots, \sigma'_j, \dots, \sigma'_n | \sigma'_d$

$$\sigma'_i = \sigma_i \exp(t_1 N(0, 1) + t_2 N_i(0, 1)), i \in [1, n]$$

$$x'_i = x_i + N(0, \sigma'_i), i \in [1, n]$$

$$\sigma'_d = \sigma_d \exp(t_{d1} N(0, 1) + t_{d2} N_d(0, 1))$$

$$d' = d + N(0, \sigma'_d)$$

其中,  $N(0, 1)$  为对个体的标准正态分布的随机变量,  $N_i(0, 1)$  和  $N_d(0, 1)$  表示对个体下标  $i$  重新采样的标准正态分布的随机变量。变异的幅度由变异策略参数决定, 策略参数的变异由算子集参数  $t_1, t_2, t_{d1}, t_{d2}$  决定,  $t_1, t_{d1}$  是整体步长,  $t_2, t_{d2}$  是基因步长, 其中  $t_1, t_2$  与任务总数及当前个体的目标函数值

关系较大。本文对影响策略参数的相关因素进行全面分析后将策略参数取为:

$$t_1 = c / \sqrt{2n}, t_2 = c / \sqrt{2} \sqrt{2n}$$

$$t_{d1} = c' / \sqrt{2}, t_{d2} = c' / \sqrt{2} \sqrt{2}$$

其中,  $n$  为任务总数。

变异算子是进化策略中最重要的算子, 为提高算法的性能, 使用模糊逻辑动态控制算子集参数  $t_1, t_2, t_{d1}, t_{d2}$  中的参数  $c$  来调整变异的幅度。首先定义目前最优个体的目标值为  $f_{opt}$ 。若个体  $S$  的目标值为  $f$ , 则个体  $S$  与最优个体的相对距离  $d = (f - f_{opt}) / f_{opt}$ 。当  $d$  较小时, 个体  $S$  接近最优个体, 如果减小变异幅度, 可以防止大量破坏个体的优良特征, 同时进行小幅度变异搜索, 从而增强算法的局部搜索能力; 当  $d$  极小时, 个体  $S$  就近似或等于最优个体, 此时应该增大变异幅度, 使个体  $S$  跳出局部极点, 防止群体早熟, 从而增强算法的全局搜索能力; 当  $d$  较大时, 取适当的变异幅度。为实现这样的变异思想, 在个体的变异中用如下的模糊逻辑动态控制算子集参数  $t_1, t_2, t_{d1}, t_{d2}$  中的参数  $c$ :

Rule 1: If  $d$  is very small, then  $c$  is big.

Rule 2: If  $d$  is small, then  $c$  is small.

Rule 3: If  $d$  is big, then  $c$  is medium.

#### (4) 重组

随机选取两个个体, 对相应第一段向量和第三段向量分别进行重组操作, 保持其余两段向量不变。令  $S$  和  $T$  是随机选取的两个父辈个体,  $p_1$  和  $p_2$  是随机整数且  $1 \leq p_1 \leq p_2 \leq n$ , 则新个体  $N$  中第一段向量和第三段向量分别为

$$x_{N,i} = \begin{cases} x_{S,i}, & i \in [p_1, p_2] \\ x_{T,i}, & i \notin [p_1, p_2] \end{cases}, \sigma_{N,i} = \begin{cases} \sigma_{S,i}, & i \in [p_1, p_2] \\ \sigma_{T,i}, & i \notin [p_1, p_2] \end{cases}, i \in [1, n]$$

#### (5) 选择

在进化策略中, 采用确定性选择方法  $(\mu, \lambda) - ES$ 。  $(\mu, \lambda) - ES$  是从  $\lambda$  个子代个体集中选择  $\mu$  ( $1 \leq \mu \leq \lambda$ ) 个最好的个体。

#### (6) 进化精调操作

和其它进化计算方法一样, 进化策略的优势在于全局搜索能力, 而局部搜索能力较差, 考虑到本文问题的复杂性和搜索空间之大, 为了进一步提高进化策略的搜索效率, 在进化策略算法中引入了基于邻域的局部搜索机制, 即进化精调操作, 来改善进化策略算法的局部搜索能力。

使用的是单方向(朝着目标值增大的方向)连续多次的对换操作, 即对应给定的个体, 在个体第一段向量中随机地选两个位置进行对换操作, 若对换使个体的目标值提高, 则执行对换操作, 如此反复多次。这一操作实际上使给定的个体改良到它的局部极点, 可大大提高搜索效率, 从而改进了算法性能。

综上所述, 算法流程为:

$t = 0$

初始化:  $P(0) = \{a_1(0), a_2(0), \dots, a_\mu(0)\}$

目标函数计算:

$F(0) = \{f(a_1(0)), f(a_2(0)), \dots, f(a_\mu(0))\}$

while(! 结束条件) do

重组:  
 $P'(t)=\{a'_1(t), a'_2(t), \cdots, a'_\lambda(t)\}$   
变异:  
 $P''(t)=\{a''_1(t), a''_2(t), \cdots, a''_\lambda(t)\}$   
目标函数计算:  
 $F''(t)=\{f(a''_1(t)), f(a''_2(t)), \cdots, f(a''_\lambda(t))\}$   
选择:  
 $P'''(t)=\{a'''_1(t), a'''_2(t), \cdots, a'''_\mu(t)\}$   
进化精调:  
 $P(t+1)=\{a_1(t+1), a_2(t+1), \cdots, a_\mu(t+1)\}$   
 $t=t+1$   
end

4 数值计算及分析

将算法(简称 ES)用 Visual C++ 编程, 在 P II 400 上用实例进行了验证. 并与文献[ 2, 7] 提出的启发式算法和遗传算法进行了比较.

在本文的数值计算中, 对每一组  $m$  和  $n$  的组合, 加工时间(均取两位数以内的整数)用随机方法产生. 目标函数中取  $\alpha=1, \beta=2, \gamma=3$ . 在进化参数的模糊逻辑控制, 使用 product-sum 重心推理法, 相对距离  $d$  使用梯形模糊数, 算子参数  $c$  使用三角模糊数. 经过选择, 以下参数是较优的. 对  $d$ :  $very\ small=[0, 0, 0.005, 0.01]$ ,  $small=[0.005, 0.01, 0.08, 0.1]$ ,  $big=[0.08, 0.15, 1.0, 1.0]$ . 对  $c$ :  $small=[0.0008, 0.001, 0.002]$ ,  $medium=[0.8, 1, 2]$ ,  $big=[8, 10, 12]$ . 种群大小  $N=50$ , 进化代数  $G=60, \mu/\lambda=1/7$ .

产生 11 个不同规模的问题, 对每个问题 ES 程序运行 20 次, 比较每个问题的目标函数, 结果如表 1 所示, 其中  $f_{min}, f_{max}$  和  $f_{avg}$  分别为目标函数  $f$  的最小计算值, 最大计算值和平均计算值. 比较每个问题的公共交货期, 结果如表 2 所示. 可以看出, 用进化策略算法所得解的性能明显优于启发式算法, 所需的计算时间也很短, 且能得到满足要求的多组解.

表 1 数值计算结果—目标函数

规模 $n * m$	启发 算法	进化策略				
		$f_{min}$	$f_{max}$	$f_{avg}$	目标函数 方差	运行时间(s)
20 * 4	1120	1110	1110	1110	0	5
20 * 6	805	777	779	777.3	0.7327	6
30 * 4	2745	2699	2707	2701.65	2.2070	9
30 * 6	1673	1653	1653	1653	0	9
30 * 8	1159	1102	1107	1102.45	1.2344	10
40 * 4	3621	3572	3577	3573	1.2566	12
40 * 6	2891	2856	2861	2857.05	1.7313	13
40 * 8	2062	2022	2024	2022.1	0.4472	13
60 * 4	10594	10589	10592	10589.2	0.8208	21
60 * 6	7723	7713	7713	7713	0	22
60 * 8	5636	5595	5616	5602.95	5.9910	21

表 2 数值计算结果—公共交货期

规模 $n * m$	启发 算法	进化策略		
		最小计算值	最大计算值	平均计算值
20 * 4	23	23	26	24
20 * 6	17	18	19	18.85
30 * 4	36	34	39	36.95
30 * 6	23	23	24	23.6
30 * 8	16	15	17	15.95
40 * 4	32	32	36	33.9
40 * 6	28	27	29	28
40 * 8	21	21	23	21.4
60 * 4	63	63	68	65.1
60 * 6	48	47	51	49.35
60 * 8	37	35	39	37

对上述各问题, 在加工时间上叠加 10% 幅度的扰动, 每个问题 ES 程序运行 20 次. 计算结果如表 3 所示, 其中  $f_{avg}$  和  $f'_{avg}$  分别为扰动前后目标函数  $f$  的平均计算值. 可以看出, 进化策略算法对加工时间的微小变化不敏感.

表 3 加工时间叠加 10% 扰动后的目标函数平均值计算结果

规模 $n * m$	$f_{avg}$	$f'_{avg}$	$\frac{ f_{avg}-f'_{avg} }{f_{avg}}$
20 * 4	1110	1092	1.62%
20 * 6	777.3	775.9	0.18%
30 * 4	2701.65	2744.05	1.57%
30 * 6	1653	1666	0.79%
30 * 8	1102.45	1093.35	0.83%
40 * 4	3573	3531.9	1.15%
40 * 6	2857.05	2833.35	0.83%
40 * 8	2022.1	1991.85	1.50%
60 * 4	10589.2	10662.1	0.69%
60 * 6	7713	7641	0.93%
60 * 8	5602.95	5543.1	1.07%

进化策略与遗传算法同属进化算法, 使用文献[ 7] 提出的遗传算法与进化策略比较, 结果如表 4 所示, 可以看出进化策略的结果优于遗传算法.

表 4 进化策略与遗传算法的目标函数均值比较

规模 ( $n * m$ )	遗传算法	进化策略
20 * 4	1113	1110
20 * 6	782	777.3
30 * 4	2713	2701.7
30 * 6	1661	1653
30 * 8	1117	1102.4
40 * 4	3584	3573
40 * 6	2874	2857
40 * 8	2040	2022.1

进化策略是一种概率搜索算法, 问题本身也有随机性, 为避免偶然结果, 每种规模生成 20 个问题, 每个问题 ES 程序运行 20 次, 比较每种规模的目标函数均值, 结果如表 5 所示. 从统计意义上看, 进化策略算法得到的解的性能仍然优于启发式算法.

表 5 数值计算结果

规模( $n * m$ )	启发式算法	进化策略
20 * 4	1303.05	1301.3
20 * 6	935.55	917.11
30 * 4	2862	2847.63
30 * 6	1983.85	1980.105
30 * 8	1567.75	1535.815
40 * 4	4956.35	4954.89
40 * 6	3429.7	3408.3
40 * 8	2642.05	2637.55
60 * 4	11014	11012.8
60 * 6	7513.75	7511.943
60 * 8	5780.6	5754.33

5 结论

本文深入研究了进化策略在带公共交货期的并行机器生产线提前/拖后调度问题上的应用. 实数编码和整数编码相结合的复合编码方式为进化策略的使用带来极大的方便. 运用模糊逻辑动态控制进化策略参数, 同时提高了算法的全局搜索和局部搜索能力. 进化精调操作进一步提高了算法的局部搜索能力. 计算实例表明本文提出的进化策略算法有很好的效果.

参考文献:

[ 1 ] 陈国良, 王煦法, 等. 遗传算法及其应用 [ M ]. 人民邮电出版社. 1996, 6: 271—284.

[ 2 ] T C E Cheng. A heuristic for common due-date assignment and job scheduling on parallel machines [ J ]. Journal of the Operational Research Society, 1989, 40( 12): 1129—1135.

[ 3 ] Bahram Alidaee, Duane Rosa. Scheduling parallel machines to minimize total weighted and unweighted tardiness [ J ]. Computers & Operations Research, 1997, 24(8): 775—788.

[ 4 ] C Koulamas G J Kyparisis. Scheduling on uniform parallel machines to minimize maximum lateness [ J ]. Operations Research Letters, 2000, 26(4): 175—179.

[ 5 ] Sivrikaya-Serifoglu Funda, Ulusoy Gündüza. Parallel machine scheduling with earliness and tardiness penalties [ J ]. Computers & Operations Research, 1999, 26(8): 773—787.

[ 6 ] Grefenstette J J. Optimization of Control Parameters for Genetic Algorithms [ J ]. IEEE Transactions on Systems, Man and Cybernetics, 1986, VSMC16(1): 122—128.

[ 7 ] 刘民, 吴澄, 张龙. 确定并行多机调度问题的公共交货期和最优调度 [ J ]. 清华大学学报, 1999, 9(39): 114—117.

[ 8 ] 刘民, 吴澄, 戴元顺. 最小化拖期任务数并行机调度问题的一种基于知识的遗传算法 [ J ]. 电子学报, 1999, 27(9): 130—132.

作者简介:



杨英杰 男, 1976 年生于新疆石河子市. 清华大学自动化系国家 CIMS 工程技术中心博士生. 主要研究领域包括: 复杂系统智能优化, 生产调度, 进化计算等.

刘 民 男, 1965 年生于浙江省杭州市. 清华大学自动化系国家 CIMS 工程技术中心讲师、博士, 中国自动化学会名词工作委员会副主任委员. 目前感兴趣的领域为复杂制造系统智能优化理论与方法、人工生命、进化计算等.