

求解不相关并行机调度的一种自适应分布估计算法

吴楚格, 王凌, 郑晓龙

(清华大学自动化系, 北京 100084)

摘要: 针对不相关并行机调度问题, 提出一种基于信息熵的自适应分布估计算法. 根据问题特性, 设计了面向工件机器分配的概率模型及其基于增量学习的更新方式, 学习速率基于信息熵进行调整. 为了增强算法局部寻优能力, 采用基于关键机器的邻域结构进行局部搜索; 同时讨论了信息熵与学习速率的关系, 并探讨了关键参数对算法性能的影响. 基于标准算例的测试结果与算法比较, 验证了学习速率的自适应调整机制以及所提出算法的有效性.

关键词: 不相关并行机; 分布估计算法; 自适应机制; 信息熵

中图分类号: TP18

文献标志码: A

An adaptive estimation of distribution algorithm for solving the unrelated parallel machine scheduling

WU Chu-ge, WANG Ling, ZHENG Xiao-long

(Department of Automation, Tsinghua University, Beijing 100084, China. Correspondent: WANG Ling, E-mail: wangling@mail.tsinghua.edu.cn)

Abstract: An entropy-based adaptive estimation of the distribution algorithm(AEDA) is proposed to solve the unrelated parallel machine scheduling problem. According to the characteristic of the problem, a job-machine assignment oriented probabilistic model and its incremental learning based updating method are designed. The learning rate is adjusted with the guidance of the information entropy. To enhance the local exploitation ability, a neighborhood structure based on the critical machine is used for local search. Moreover, the relation between information entropy and learning rate is discussed, and the effect of key parameters on the performance of the algorithm is investigated. Testing results and the comparisons to the existing algorithms by using the benchmark instances demonstrate the effectiveness of both the adaptive adjusting mechanism of the learning rate and the proposed algorithm.

Keywords: unrelated parallel machine; estimation of distribution algorithm; adaptive mechanism; information entropy

0 引言

并行机调度是实际生产制造过程中的一类典型调度问题^[1]. 不相关并行机调度问题(UPMSP)是并行机调度最普遍的一类问题, 工件的加工时间取决于所分配的机器. UPMSP具有广泛的应用背景, 譬如纺织制造^[2]、半导体加工^[3]、云计算^[4]等. 数学上, UPMSP已被证明为NP-hard问题^[1]. 大规模UPMSP的求解相当困难, 其高效算法的研究一直是相关领域的重要课题. 早期算法主要是精确求解方法和调度规则, 包括动态规划^[5]、分枝定界^[6]和启发式规则^[7]. 精确求解方法所需计算量和存储空间均较大, 难以高效求解大规模问题. 启发式规则虽然易于实施, 但所得解的质量一般不高. 对于大规模问题, 近些年智能算法

及其与启发式规则的混合算法得到了广泛关注. 譬如, 两阶段蚁群算法^[8]、基于贪婪迭代的局部搜索方法^[9-10]、遗传算法^[11]、禁忌搜索及其与分枝定界的混合算法^[11].

分布估计算法(EDA)^[12]是一种基于统计学原理的群体进化算法. EDA采用概率模型描述解空间, 通过采样概率模型产生新个体, 并基于优势个体的信息更新概率模型. 概率模型是EDA的关键. 对于最小化最大完成时间的UPMSP, 在不考虑工件的加工设置时间的前提下, 求解问题的关键就是要解决工件-机器的合理分配. 为了使用EDA有效求解UPMSP, 本文采用变量不相关的概率模型描述工件-机器的分配关系, 并利用信息熵自适应调整模型的学习速率, 同时

收稿日期: 2015-10-13; 修回日期: 2015-12-17.

基金项目: 国家杰出青年科学基金项目(61525304); 高等学校博士学科点专项科研基金项目(20130002110057).

作者简介: 吴楚格(1993-), 女, 博士生, 从事智能优化调度的研究; 王凌(1972-), 男, 教授, 博士生导师, 从事智能优化调度理论与方法等研究.

采用基于关键机器的局部搜索加强局部搜索能力,进而提出一种基于信息熵的自适应EDA(AEDA).

1 不相关并行机调度问题

UPMSP考虑 n 个工件在 m 台不相关并行机上的加工过程,假设不考虑工件的加工设置时间,调度目标是通过工件-机器的合理分配来最小化最大完成时间. $R||C_{\max}^{[13]}$ 的数学模型描述如下:

$$\min C_{\max}; \quad (1)$$

$$\sum_{i=1}^m x_{ij} = 1, j = 1, 2, \dots, n; \quad (2)$$

$$\sum_{j=1}^n p_{ij}x_{ij} \leq C_{\max}, i = 1, 2, \dots, m; \quad (3)$$

$$x_{ij} \in \{0, 1\}, i = 1, 2, \dots, m, j = 1, 2, \dots, n. \quad (4)$$

其中: $x_{ij} = 1$ 表示将工件 j 分配至机器 i , C_{\max} 表示工件的最大完成时间,工件加工时间 p_{ij} 取决于所分配的加工机器.

2 自适应分布估计算法

本节首先介绍基本EDA的流程,然后给出自适应EDA的编码、概率模型的初始化与更新机制、局部搜索操作,最后给出自适应EDA的流程.

2.1 EDA

基于统计学习的原理,EDA根据问题信息建立概率模型来模拟解空间中个体的分布,进而通过对模型的随机采样产生新种群,并利用优势个体的信息更新模型,如此反复,进而实现种群的进化.

标准分布估计算法的步骤如下:

Step 1: 初始化概率模型;

Step 2: 随机采样概率模型产生新种群;

Step 3: 确定种群中的优势个体;

Step 4: 利用优势个体更新概率模型;

Step 5: 判断终止条件是否满足,若满足,则输出最优解,否则返回Step 2.

2.2 编码方式

鉴于UPMSP的特点,采用针对工件的排列编码方式,即 $[m_1, m_2, \dots, m_i, \dots, m_n]$,其中 m_i 表示工件 i 被分配的加工机器.该编码方式简单、直观.

2.3 概率模型的初始化

AEDA采用变量不相关模型,即概率矩阵 $Q = [q_{ij}]$ 来指导种群的产生,其中 $q_{ij} = p(x_{ij})$ 表示工件 j 分配到机器 i 加工的概率.显然 $\sum_{i=1}^m q_{ij} = 1$.初始化模型时,采用如下均匀分布方式以实现对解空间的均匀抽样:

$$q_{ij} = \frac{1}{m}, i = 1, 2, \dots, m, j = 1, 2, \dots, n. \quad (5)$$

2.4 概率模型的更新机制

EDA的核心搜索操作是基于概率矩阵采样产生新个体.对于每个编码个体的第 j ($j = 1, 2, \dots, n$)位,根据 q_{ij} ($i = 1, 2, \dots, m$)利用轮盘赌的方式确定机器编号.一旦确定了所有位上的机器,就确定了该个体的具体加工情况,进而可评价其最大完成时间.按上述方式产生 M 个个体构成当前种群.对于当前种群,选择目标值最好的 N 个个体($N < M$)作为优势个体,进而采用群体增量学习(PBIL)^[14]方式更新概率矩阵

$$q_{ij}^{l+1} = (1 - \alpha)q_{ij}^l + \alpha \frac{1}{N} \sum_{k=1}^N \delta_l^k(i, j). \quad (6)$$

其中: q_{ij}^l 为第 l 代中工件 j 分配到机器 i 的概率, α ($0 < \alpha < 1$)为学习速率.若 l 代第 k 个优势个体 $m_i = j$,则 $\delta_l^k(i, j) = 1$,否则, $\delta_l^k(i, j) = 0$.

2.5 基于关键机器的局部搜索

改善智能算法性能的关键之一是要增强并均衡全局探索与局部趋化能力.EDA基于概率模型进行全空间的采样,全局搜索能力较强.为了增强局部趋化搜索能力,采用针对问题的特定局部搜索操作进行深度搜索.针对UMPSP的特点,采用交换邻域结构和基于关键机器的插入邻域结构^[9]进行局部搜索操作,其中关键机器指完成时间最大的机器.

交换操作 两两交换所有工件的加工机器,每次交换后,若两个工件的加工时间之和减小且 C_{\max} 不增加,则接受加工机器的交换.交换搜索的伪代码如下:

```

for i = 1; i ≤ n; i ++ do
  for j = i; j ≤ n; j ++ do
    if p(i, mi) + p(j, mj) > p(i, mj) + p(j, mi)
      and C'_{max} ≤ C_{max} then
        mi ← j;
        mj ← i;
        break;
  end
end
end

```

其中 $p(j, M_j)$ 表示工件 j 分配到机器 M_j 的加工时间, C_i 表示机器 i 的加工完成时间.

插入操作 对于关键机器,顺序选取其加工的工件,考虑将该工件顺序移到其他机器上加工,若能减小 C_{\max} ,则接受移动.插入搜索的伪代码如下:

```
times = 0;
```

```

while times < LStime do
  for  $j = \arg(m_j = \text{Num})$  do
    times++;
     $i^* = \operatorname{argmin}(C_i + p(j, i)), i = 1, 2, \dots, m;$ 
    if  $C_{i^*} + p(j, i^*) < C_{\max}$  then
       $m_j \leftarrow i^*;$ 
      calculate Num;
    end
  end
end
end

```

其中局部搜索深度 LStime 设置为 50.

2.6 学习速率的自适应调整机制

研究表明^[15], EDA 的学习速率过大会导致算法早熟收敛, 而过小会导致算法收敛缓慢. 在此, 设计学习速率的一种基于信息熵的自适应调整机制.

信息熵是信息论中衡量随机变量出现的期望值和测量变量不确定度的一个度量^[16]. EDA 概率模型的信息熵可以反映概率模型的无序程度, 表征基于概率模型采样所得种群的多样性, 可用于调整学习速率. 当概率模型信息量较大时学习速率较大, 有利于快速的全局搜索. 随着种群进化, 概率模型的信息量降低, 学习速率减小, 有利于细致的局部搜索.

随机变量 X 的信息熵计算如下:

$$H(X) = -\sum_i p(x_i) \log_p(p(x_i)). \quad (7)$$

其中: x_i 为变量 X 的可能取值, $p(x_i)$ 为 x_i 的出现概率, p 为信息量对数的底, 通常设为 2.

对于相互独立的多个变量, 其联合信息熵计算如下:

$$H(X_1, X_2, \dots, X_i, \dots, X_n) = -\sum_i \sum_j p(x_{ij}) \log_p(p(x_{ij})), \quad (8)$$

其中 $p(x_{ij})$ 表示第 i 个变量的第 j 种取值的概率.

对于 AEDA 的概率模型 $Q = [q_{ij}]$, 记工件 j 所分配的机器为随机变量 M_j , 根据式 (7), M_j 的信息熵为

$$H(M_j) = -\sum_{i=1}^m q_{ij} \log_p(q_{ij}). \quad (9)$$

对于每个个体, 鉴于工件间的独立性, 根据式 (8), 所有工件的机器分配的联合信息熵为

$$H(M_1, M_2, \dots, M_j, \dots, M_n) = -\sum_{j=1}^n \sum_{i=1}^m q_{ij} \log_p(q_{ij}). \quad (10)$$

考虑概率模型基于增量学习的更新方式, 即式 (6), 令

$$r_{ij}^l = \frac{1}{N} \sum_{k=1}^N \delta_l^k(i, j),$$

则第 $l+1$ 代中工件 j 分配到机器 i 的概率可改写为

$$q_{ij}^{l+1} = (1 - \alpha)q_{ij}^l + \alpha r_{ij}^l.$$

进而, 令 $f(x) = -x \log(x)$, 则第 $l+1$ 代中工件 j 所分配的机器的信息熵为

$$H(M_j)^{l+1} = \sum_{i=1}^m f((1 - \alpha)q_{ij}^l + \alpha r_{ij}^l). \quad (11)$$

其中: $\sum_{i=1}^m q_{ij} = 1, \sum_{i=1}^m r_{ij} = 1.$

对于基于增量学习的更新方式, α 较小时, 概率模型较大程度保持上一代信息, 信息熵变化缓慢; 反之, 概率模型较大程度依赖优势个体进行调整, 信息熵变化较快. 对于 EDA 算法而言, 种群进化初始时, 信息熵相对较大, 设置较大的 α 使得概率模型可根据优势个体快速得到调整, 有利于算法在解空间快速趋于性能较好的区域; 在种群进化后期, 信息熵相对较小, 设置较小的 α 使得概率模型变化缓慢, 有利于算法在较好区域进行细搜索. 对于前文设计的初始化概率模型, 相应的信息熵为

$$H^0 = -\sum_{j=1}^n \sum_{i=1}^m \frac{1}{m} \log_p \frac{1}{m} = n \log_p m. \quad (12)$$

因此, 设计学习速率的自适应调整机制如下所示:

$$\alpha^l = \alpha^0 \exp(H^l / (n \log_p m) - 1), \quad (13)$$

使得学习速率与信息熵正相关. 其中 α^0 为初始学习速率.

下面通过仿真来验证上述分析. 考虑标准测试集 (<http://soa.iti.es/problem-instances>) 中一个 $n = 100$ 、 $m = 10$ 的测试问题. 设置 $M = 200$, $N = 5$, α 分别取为 0.1、0.06 和 0.03, 对算法分别运行 10s, 则第 1 号工件的信息熵随进化代数的变化曲线如图 1 所示. 由图 1 可见, 学习速率大时, 信息熵下降较快; 反之, 信息熵下降较慢. 仿真结果与前文分析一致.

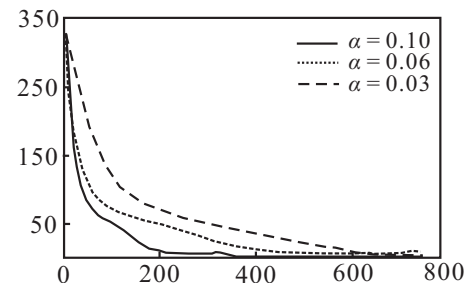


图 1 不同学习速率下单信息熵曲线

2.7 算法流程

鉴于上述设计, 给出求解 UPMSPP 的自适应分布估计算法流程, 如图 2 所示. 相对传统 EDA 的流程, AEDA 增加的局部搜索环节及根据信息熵自适应调整概率模型的学习速率, 有助于增强算法的搜索能力.

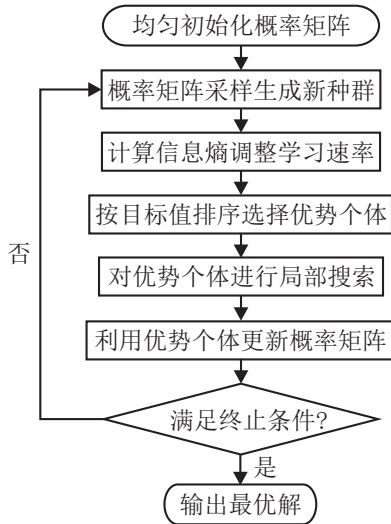


图 2 AEDA 流程图

3 仿真结果比较

下面利用不相关并行机调度问题的标准测试集 $U(1, 100)$ (<http://soa.iti.es/problem-instances>) 进行算法性能测试与比较. 该测试集包括 20 种规模的问题, 工件数 $n = \{100, 200, 500, 1000\}$, 机器数 $m = \{10, 20, 30, 40, 50\}$, 工件加工时间为 1~100 均匀分布的随机整数, 每种规模含 10 个测例, 共 200 个测例. 仿真计算机 CPU 主频 2.83 GHz, 内存 4 GB, 采用 C 语言编制程序并在 Windows 7 系统运行. 下面首先探讨参数对算法性能的影响, 然后给出性能测试与算法比较结果.

3.1 算法参数设置

AEDA 的主要参数包括种群规模 M , 优势个体数 N 和初始学习速率 α^0 . 采用第 241 号测例 ($m = 40, n = 200$) 开展实验设计 (DOE)^[17]. 每个参数设 4 个水平, 如表 1 所示. 根据正交表 $L_{16}(4^3)$ 对每组参数设置独立运行算法 10 次, 每次运行 10 s, 算法所得平均性能作为响应变量 (RV), 结果如表 2 所示. 进而, 根据表 2 计算得到各参数的响应值以及参数对算法性能的影响等级, 结果如表 3 所示, 各参数对算法性能的影响趋势如图 3 所示.

表 1 AEDA 参数各水平取值

参数	水平			
	1	2	3	4
M	100	200	300	400
N	3	5	7	9
α^0	0.02	0.06	0.10	0.14

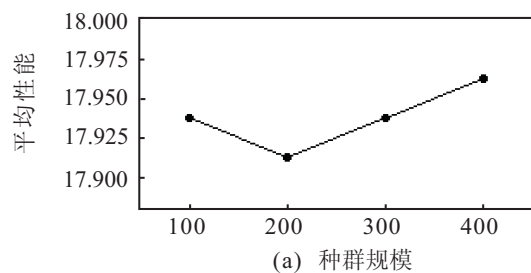
由表 3 可见, 优势个体数对算法性能影响最大, 其次是初始学习速率, 最后是种群规模. 由图 3 可见, 3 个参数均不宜太大或太小. 对于优势个体数而言, 太大则使算法过度侧重于粗搜索而收敛缓慢, 太小则使算法受制于少量个体而易出现早熟收敛. 对于初始学习速率而言, 初始速率太大则使得概率模型的调整幅

表 2 参数正交表及其 RV 值

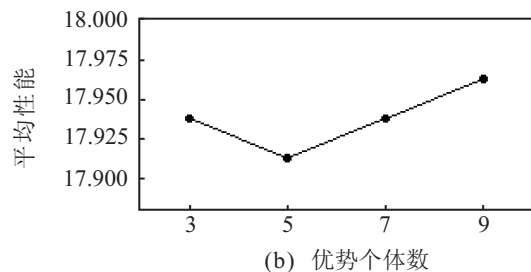
组合编号	水平			RV
	M	N	α^0	
1	1	1	1	18.00
2	1	2	2	18.00
3	1	3	3	17.80
4	1	4	4	17.95
5	2	1	2	18.00
6	2	2	1	17.85
7	2	3	4	17.90
8	2	4	3	17.90
9	3	1	3	18.00
10	3	2	4	17.80
11	3	3	1	17.95
12	3	4	2	18.00
13	4	1	4	18.00
14	4	2	3	17.90
15	4	3	2	17.95
16	4	4	1	18.00

表 3 AEDA 各参数平均 RV 值

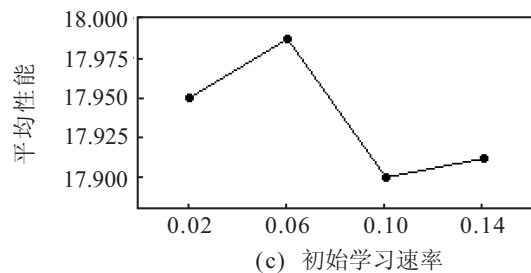
水平	M	N	α^0
1	17.94	18.00	17.95
2	17.91	17.89	17.99
3	17.94	17.90	17.90
4	17.96	17.96	17.91
极差	0.05	0.11	0.09
等级	3	1	2



(a) 种群规模



(b) 优势个体数



(c) 初始学习速率

图 3 参数对算法性能影响的水平趋势

度较大而导致细搜索能力不强, 太小则使模型的调整幅度较小而导致算法全局搜索能力不足. 对于种群规模而言, 在总计算时间固定的前提下, 种群规模太大则使算法的总进化代数变小而导致进化不充分, 太小则使算法每代的群体进化能力变弱. 基于上述探讨, 在后续实验中 AEDA 设置 $M = 200, N = 5, \alpha^0 = 0.1$.

3.2 学习速率自适应调整机制有效性验证

为验证学习速率的自适应调整机制的有效性, 将 AEDA 与固定学习速率的 EDA ($\alpha = 0.03$) 进行比较, 两算法对每个测例均独立运行 10 次, 每次运行 10 s, 进而将相同规模的 10 个测例的 100 次结果作为一组进行统计检验. 考虑到各次运行结果未必服从正态分布, 在 95% 置信水平下采用非参数检验方法曼-惠特尼检验 (Mann-Whitney U test)^[18] 检验两种算法的性能是否存在显著差异. 结果如表 4 所示, 其中 η_{AEDA} 和 η_{EDA} 表示两算法所得结果的中位数. 若检验统计量小于 0.05, 则表明 AEDA 的性能显著优于 EDA.

表 4 学习速率自适应调整机制有效性验证

n, m	η_{AEDA}	η_{EDA}	检验统计量	是否显著
100, 10	105.0	105.0	—	N
100, 20	32.0	32.0	—	N
100, 30	18.5	18.5	—	N
100, 40	13.0	13.0	0.485 5	N
100, 50	10.0	10.0	—	N
200, 10	200.0	200.0	—	N
200, 20	59.0	59.0	0.140 7	N
200, 30	30.0	30.0	0.005 6	N
200, 40	19.0	20.0	0.000 0	Y
200, 50	14.0	15.0	0.000 0	Y
500, 10	491.5	492.5	0.101 9	N
500, 20	143.0	145.0	0.007 0	Y
500, 30	73.0	77.0	0.000 0	Y
500, 40	46.0	48.0	0.000 0	Y
500, 50	33.0	36.0	0.000 0	Y
1000,10	980.5	982.5	0.019 4	Y
1000,20	280.0	285.0	0.000 0	Y
1000,30	140.0	145.0	0.000 0	Y
1000,40	88.0	93.0	0.000 0	Y
1000,50	64.0	67.0	0.000 0	Y

由表 4 可见, 算法在自适应变速率机制的性能基本都优于固定速率算法, 尤其在大规模问题上表现出显著的优越性. 因此, 学习速率的自适应调整机制有助于改善 EDA 的性能.

另外, 采用一个 $n = 100, m = 10$ 的测试问题, 给出学习速率分别为 0.1、0.03 和自适应机制 (初始速率为 0.1) 下的算法所得最优性能随进化代数的变化曲线, 如图 4 所示. 曲线再次验证了自适应变速率机制在收敛质量与效率方面的优越性.

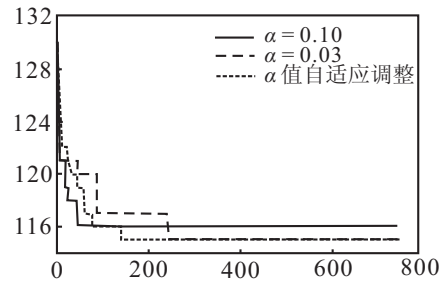


图 4 不同学习速率下最优个体性能的下曲线

3.3 仿真结果比较与分析

下面, 将 AEDA 与并行遗传算法 (PGA)^[19] 进行比较. 另外, 由于文献 [19] 采用 10 台服务器运行算法, 本文给予 PGA 100 s 与 AEDA (运行 10 s) 在同一计算机上进行性能比较. 对每个测例均独立运行算法 20 次, 以所得最优解和平均解相对 C_{max}^* 的百分偏差

$$RPD = \frac{100(C_{max} - C_{max}^*)}{C_{max}^*} \quad (14)$$

作为性能评价指标, 其中 C_{max}^* 为 CPLEX11.0 运行 2 h 所得的最优结果, 对比结果如表 5 所示.

表 5 AEDA 与 PGA 结果比较

n, m	AEDA		PGA		是否显著
	最优偏差	平均偏差	最优偏差	平均偏差	
100, 10	1.06	1.75	3.74	6.61	Y
100, 20	2.49	4.23	6.64	11.39	Y
100, 30	1.10	2.36	6.07	9.06	Y
100, 40	3.58	5.32	12.29	16.94	Y
100, 50	1.25	2.99	9.10	14.44	Y
200, 10	0.99	1.34	4.80	7.07	Y
200, 20	2.98	4.07	10.42	14.48	Y
200, 30	5.46	7.42	14.33	18.32	Y
200, 40	7.74	10.42	17.06	23.17	Y
200, 50	12.12	14.02	22.36	28.44	Y
500, 10	0.93	1.15	6.94	8.82	Y
500, 20	4.96	5.83	13.11	15.86	Y
500, 30	10.75	12.31	17.50	20.54	Y
500, 40	15.73	17.70	19.34	22.13	Y
500, 50	23.62	26.56	23.23	2687	N
1000,10	0.99	1.19	9.55	10.97	Y
1000,20	4.47	4.95	14.75	16.52	Y
1000,30	8.77	10.21	20.06	21.70	Y
1000,40	17.58	19.79	21.79	23.36	Y
1000,50	23.03	24.57	24.57	25.93	Y
平均值	7.48	8.09	13.88	17.17	—

由表 5 可见, AEDA 在所有测试问题上的平均性能均优于 PGA, 在最优性能上除 (500,50) 问题外均优于 PGA. 因此, AEDA 是求解 UPMS 的一种更有效的算法. AEDA 性能的优越性主要归功于两方面, 一是基于概率模型的全局搜索与基于关键机器的局部搜索的结合; 二是基于信息熵的学习速率自适应调整.

4 结 论

本文提出了求解不相关并行机调度问题的一种自适应分布估计算法. 一方面, 将基于关键机器的局部搜索引入算法, 均衡了算法的全局搜索与局部搜索能力; 另一方面, 通过引入基于信息熵的学习速率自适应调整机制, 使得算法的搜索能力得到动态调整. 仿真比较与分析验证了所提出自适应策略和算法的有效性. 进一步的工作可探讨优势个体数与种群数的自适应调整机制, 并探讨自适应分布估计算法在其他类型的生产调度问题(包括多目标问题)上的应用.

参考文献(References)

- [1] 王凌. 车间调度及其遗传算法[M]. 北京: 清华大学出版社, 2003: 146-148.
(Wang L. Shop scheduling with genetic algorithms[M]. Beijing: Tsinghua University Press, 2003: 146-148.)
- [2] Kerkhove L P, Vanhoucke M. Scheduling of unrelated parallel machines with limited server availability on multiple production locations: A case study in knitted fabrics[J]. *Int J of Production Research*, 2014, 52(9): 2630-2653.
- [3] 张刚志, 周炳海. 带换模约束的非等效并行机作业族调度方法[J]. *计算机集成制造系统*, 2013, 19(12): 3162-3169.
(Zhang G Z, Zhou B H. Job families scheduling for non-identical parallel machines with setup time constraints[J]. *Computer Integrated Manufacturing Systems*, 2013, 19(12): 3162-3169.)
- [4] Hassan A M A, Kacem I, Martin S. Unrelated parallel machines with precedence constraints: Application to cloud computing[C]. *IEEE 3rd Int Conf on Cloud Networking*. Luxembourg, 2014: 438-442.
- [5] Horowitz E, Sahni S. Exact and approximate algorithms for scheduling nonidentical processors[J]. *J of the ACM*, 1976, 23(2): 317-327.
- [6] Martello S, Soumis F, Toth P. Exact and approximation algorithms for makespan minimization on unrelated parallel machines[J]. *Discrete Applied Mathematics*, 1997, 75(2): 169-188.
- [7] Ibarra O H, Kim C E. Heuristic algorithms for scheduling independent tasks on nonidentical processors[J]. *J of the ACM*, 1977, 24(2): 280-289.
- [8] Arnaout J, Musa R, Rabadi G. A two-stage ant colony optimization algorithm to minimize the makespan on unrelated parallel machines, part II: Enhancements and experimentations[J]. *J of Intelligent Manufacturing*, 2014, 25(1): 43-53.
- [9] Fanjul-Peyro L, Ruiz R. Iterated greedy local search methods for unrelated parallel machine scheduling[J]. *European J of Operational Research*, 2010, 207(1): 55-69.
- [10] Rodriguez F J, Lozano M, Blum C, et al. An iterated greedy algorithm for the large-scale unrelated parallel machines scheduling problem[J]. *Computers & Operations Research*, 2013, 40(7): 1829-1841.
- [11] Sels V, Coelho J, Dias A M, et al. Hybrid tabu search and a truncated branch-and-bound for the unrelated parallel machine scheduling problem[J]. *Computers & Operations Research*, 2015, 53(1): 107-117.
- [12] Mühlenbein H, Paass G. From recombination of genes to the estimation of distributions I: Binary parameters[J]. *Lecture Notes in Computer Science*, 1996, 1141(1): 178-187.
- [13] Graham R L, Lawler E L, Lenstra J K, et al. Optimization and approximation in deterministic sequencing and scheduling: A survey[J]. *Annals of Discrete Mathematics*, 1979, 5(1): 287-326.
- [14] Baluja S. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning[R]. Pittsburgh: Carnegie Mellon University, Department of Computer Science, 1994: 1-20.
- [15] 王凌, 王圣尧, 方晨. 一种求解多维背包问题的混合分布估计算法[J]. *控制与决策*, 2011, 26(8): 1121-1125.
(Wang L, Wang S Y, Fang C. A hybrid distribution estimation algorithm for solving multidimensional knapsack problem[J]. *Control and Decision*, 2011, 26(8): 1121-1125.)
- [16] Shannon C E. A mathematical theory of communication[J]. *ACM Sigmoble Mobile Computing and Communications Review*, 2001, 5(1): 3-55.
- [17] Onyiah L C. Design and analysis of experiments[J]. *Handbook of Statistics*, 2001, 30(2): 241-242.
- [18] Hollander M, Wolfe D A, Chicken E. Nonparametric statistical methods[M]. New York: John Wiley & Sons, 2013: 1-100.
- [19] 高家全, 方蕾. 解非等同并行多机调度问题的并行遗传算法[J]. *计算机工程*, 2007, 33(1): 198-199.
(Gao J Q, Fang L. Parallel genetic algorithm for solving non-identical parallel multi-machine schedule problem[J]. *Computer Engineering*, 2007, 33(1): 198-199.)