第 22 卷 第 10 期 控 制 与 决 策 2007 年 10 月
Vol. 22 No. 10 Control and Decision Oct. 2007

文章编号: 1001-0920(2007)10-1151-04

加工时间不确定的 Just-in-time 单机鲁棒调度

刘 琳, 谷寒雨, 席裕庚
(上海交通大学 自动化研究所, 上海 200240)

摘 要: 针对单机 Just-in-time 系统在加工时间有较大范围不确定性的调度环境,设计出一种两层协同进化遗传算法,解决绝对鲁棒调度优化问题,最优化在加工时间变化范围内预调度的最差性能. 外层遗传算法确定工件的加工顺序,内层遗传算法确定在给定调度顺序下实现最差性能的加工时间. 通过对大量随机算例进行仿真,并与采用期望加工时间的确定性调度算法进行对比,表明所提出的算法是有效的.

关键词: 遗传算法; 鲁棒调度; 最差性能
中图分类号: TP273 **文献标识码**: A

Robust scheduling in a Just-in-time single machine system with processing time uncertainty

LIU Lin, GU Han-yu, XI Yu-geng
(Institute of Automation, Shanghai Jiaotong University, Shanghai 200240, China. Correspondent: LIU Lin, E-mail: liulin2003@sjtu.edu.cn)

Abstract: A two-loop co-evolutionary genetic algorithm is proposed to solve the absolute robust scheduling for a Just-in-time single machine with significant processing time uncertainty. The worst-case performance of a predictive schedule over the range of job processing times is optimized. The outer loop of the proposed algorithm is to determine the job sequence on machine and the inner loop searches for the processing time scenario with worst-case performance for a given sequence. The simulation results show the proposed method is very effective compared with the deterministic scheduling method based on expected job processing times.

Key words: Genetic algorithm; Robust scheduling; Worst-case performance

1 引 言

调度问题是在时域上将稀有资源分配给相互竞争的任务,以优化一个或多个性能指标[1]. 以往的研究大多假设为理想的调度环境,但在实际的制造系统中,常常存在许多不确定因素,如工件加工时间变化、机器故障或客户更改交货期等. 因此,确定性调度很难保证在调度环境某些参数扰动下的系统性能. 鲁棒调度方法以最优化最差情况下的性能为目标,保证调度决策在所有参数下的性能都较好,对于存在较大范围扰动的调度环境是一种有效的方法[2-4].

在过去十几年中,调度的鲁棒性问题得到了广泛的研究[2-6]. 该研究领域有 3 种鲁棒指标:绝对鲁棒指标、偏差鲁棒指标和相对鲁棒指标[7]. 绝对鲁棒指标是最优化最差参数情况下的性能,后两种指标是最优化在给定参数下与最优调度性能的偏差. 文献[2,3]采用后两种指标,分别研究了单机 $1\,\|\,\sum c_i$ 和两机 flowshop $F_2\,\|\,C_{\max}$ 在加工时间不确定情况下的鲁棒调度问题,根据所研究问题的特性提出相应的优势准则,并设计了保证鲁棒性的分支定界方法和启发式方法,改善了调度在加工时间不确定环境下的鲁棒性. 文献[5,6]从最优调度的不变性角度研究调度的鲁棒性问题,定义了最优调度的区间摄动鲁棒性. 与偏差相关的鲁棒指标需要求解特定参数下的最优性能,对于 NP-hard 问题,求解最优调度本身就是很难的.

本文研究 $1\,|\,r_i\,|\,\sum_{i=1}^{n}(\alpha_i E_i + \beta_i T_i)$ 在加工时间不确定情况下的调度问题,目的是生成一个保证最差性能最优的鲁棒预调度. 采用绝对鲁棒指标,它是一

收稿日期: 2006-06-11; **修回日期**: 2006-10-08.
基金项目: 国家自然科学基金项目(60504026); 上海市科技发展基金项目(04DZ11008).
作者简介: 刘琳(1978 —),女,河北保定人,博士生,从事生产规划与调度算法等研究; 席裕庚(1946 —),男,上海人,教授,博士生导师,从事预测控制、智能机器人等研究.

个 NP-hard 问题. 通过绝对鲁棒调度优化问题的性质分析,得到了在给定顺序下,内层 max 优化问题的最优解在可行域有界凸多面体顶点的结论,大大降低了问题的搜索空间. 在此基础上,设计出一种两层协同进化遗传算法,外层确定工件的加工顺序,内层搜索在给定的加工顺序下实现最差性能的加工时间. 通过对大量算例进行仿真研究,表明该算法可得到一个保证性能界的鲁棒预调度,在加工时间变化的情况下,算法的最差性能优于以期望加工时间为基础的确定性调度,从平均性能上看与其偏差很小.

2 问题描述

2.1 原问题描述

对于单机 Just-in-time 系统,调度问题为 $1 | r_i | \sum_{i=1}^{n}(\alpha_i E_i + \beta_i T_i)$. 设有 n 个需要加工的工件 $i(i = 1, 2, \cdots, n)$,其工艺约束条件如下:

1) 任何时刻机器只能加工一个工件;

2) 工件只有在到达之后才能被加工;

3) 工件只能被加工一次;

4) 加工过程不允许抢占.

已知 $1 \| \sum_{i=1}^{n} T_i$ 是 NP-hard 问题[8], $1 | r_i | \sum_{i=1}^{n}(\alpha_i E_i + \beta_i T_i)$ 调度问题也是 NP-hard 问题. 该问题可用整数规划模型描述如下:

目标函数

$$\min \sum_{i=1}^{n}(\alpha_i E_i + \beta_i T_i); \quad (1)$$

约束条件

$$c_i \geq r_i + p_i, i = 1,2,\cdots,n, \quad (2)$$

$$c_j - c_i + M(1 - x_{ij}) \geq p_i, \quad i \neq j, i, j = 1,2,\cdots,n, \quad (3)$$

$$c_i - c_j + M x_{ij} \geq p_j, \quad i \neq j, i, j = 1,2,\cdots,n, \quad (4)$$

$$E_i = \max\{0, d_i - c_i\}, i = 1,2,\cdots,n, \quad (5)$$

$$T_i = \max\{0, c_i - d_i\}, i = 1,2,\cdots,n, \quad (6)$$

$$x_{ij} = \{0,1\}, \forall i, j, i \neq j. \quad (7)$$

其中: p_i, r_i 和 d_i 分别为工件 i 的加工时间、到达时间和交货期; α_i 和 β_i 为工件 i 的库存和拖期权重; c_i 是工件 i 的完成时间; M 是很大的正数; 决策变量

$$x_{ij} = \begin{cases} 1, \text{工件 } i \text{ 在工件 } j \text{ 前面加工;} \\ 0, \text{否则.} \end{cases}$$

式(1) 表明目标函数为工件的提前和拖期惩罚的加权和,式(2) 保证每个工件只有在到达之后才能被加工,式(3) 和(4) 说明机器在任意时段内只能加工一个工件,式(5) 和(6) 用来计算提前期和拖期,式(7) 表明决策变量为 0-1 整数变量.

2.2 绝对鲁棒问题描述

在加工时间不确定的情况下,以确定性调度算法得到的预调度难以保证在加工时间变化环境下的性能,本文的目标是生成一个在变化范围内保证鲁棒性的预调度. 这里采用的鲁棒指标是绝对鲁棒指标,它是一个 minimax 优化问题

$$\min_{x \in X}\{\max_{s \in S} \sum_{i=1}^{n} f_i^s(c_i^x)\},$$

$$\text{s.t. } (2) \sim (4), \quad (8)$$

$$f_i^s(c_i^x) = \max\{\alpha_i(d_i - c_i^x(s)), \beta_i(c_i^x(s) - d_i)\}, \quad (9)$$

$$s = [p_1, p_2, \cdots, p_n]^{\mathrm{T}}, \quad (10)$$

$$S = \{\underline{p}_i \leq p_i \leq \overline{p}_i \mid i = 1,2,\cdots,n\}. \quad (11)$$

其中: X 为所有可行调度顺序集合, c_i^x 为工件 i 在调度顺序 x 下的完成时间.

2.3 绝对鲁棒问题的性质分析

上述 minimax 问题是 NP-hard 问题,难以找到有效的多项式算法得到最优解. 下面对该问题的一些性质进行分析.

在给定顺序 x 的情况下, minimax 问题可转化为求解该顺序下的最差性能指标

$$\max_{s \in S} \sum_{i=1}^{n} f_i^s(c_i^x), \text{s.t. } (9) \sim (11). \quad (12)$$

性质 1 函数 $f(t) = \max\{\alpha_i(d_i - t), \beta_i(t - d_i)\}, t \geq 0$ 是关于 t 的分段线性凸函数.

推论 1 函数 $f_i^s(c_i^x)$ 是关于 $s = [p_1, p_2, \cdots, p_n]^{\mathrm{T}}$ 的分段线性凸函数.

证明 在给定调度顺序 x 的情况下,工件 i 的完成时间 c_i^x 是工件加工时间向量 s 的非负线性组合,且是单调增函数. 假设 $x = \{1,2,\cdots,n\}$,则工件的完成时间为

$$c_1^x = r_1 + p_1,$$

$$c_2^x = \max\{r_2, c_1^x\} + p_2,$$

$$\cdots$$

$$c_n^x = \max\{r_n, c_{n-1}^x\} + p_n.$$

显然,任意工件 i 的完成时间 c_i^x 和工件的加工时间向量 $s = [p_1, p_2, \cdots, p_n]^{\mathrm{T}}$ 都是线性且单调增的关系. 根据性质 1,可得 $f_i^s(c_i^x)$ 是关于 s 的分段线性凸函数. □

推论 2 函数 $\sum_{i=1}^{n} f_i^s(c_i^x)$ 是关于 $s = [p_1, p_2, \cdots, p_n]^{\mathrm{T}}$ 的分段线性凸函数.

证明 根据凸函数的性质,有限个凸函数的非负线性组合仍是凸函数. □

定理 1　对于任一给定顺序 x,在该顺序下取得最差性能的情况属于以加工时间为极值的集合.

证明　给定顺序 x,根据推论 2,$\sum_{i=1}^{n} f_i^s(c_i^x)$ 是关于 $s=[p_1,p_2,\cdots,p_n]^{\mathrm{T}}$ 的分段线性凸函数,且约束集 $S=\{\underline{p}_i \le p_i \le \overline{p}_i \mid i=1,2,\cdots,n\}$ 是一个凸多面体,因此 $\max_{s\in S}\sum_{i=1}^{n} f_i^s(c_i^x)$ 的极值点必在这个凸多面体的顶点取得[2]. □

3　本文设计的求解算法

近年来,遗传算法在求解调度问题特别是复杂调度问题中的应用十分广泛[9]. 对于绝对鲁棒 minimax 优化问题,本文设计出一种两层协同进化遗传算法,算法的外层确定调度顺序,内层在给定的调度顺序下搜索得到最差性能的加工时间.

3.1　遗传进化过程

(1) 染色体编码和解码

根据单机调度问题的特点,外层采用一维的自然数编码,即采用基于工件的编码. 根据到达时间的先后,对工件 1～n 进行标号. 以工件的标号表示染色体的基因,染色体长度为 n,染色体自然表征了调度的顺序. 内层采用一维的二进制编码方式,染色体长度为 n. 其中元素 0 表示对应工件的加工时间取下界值,元素 1 表示加工时间取上界值. 对于任一给定的外层调度顺序,内层遗传算法在该顺序下进化求解问题(12),其结果用来评价外层进化种群个体的适应度.

在优化原目标的同时,希望提高机器的利用率. 这里在解码过程中采用了非延迟解码方式,只要机器空闲且有工件等待便进行加工.

(2) 种群初始化

外层的初始种群由两部分组成:按照 EDD 规则和 EST 规则生成两个初始个体;然后对这两个个体进行等概率的变异操作,生成整个初始种群. 内层的初始种群直接随机生成.

(3) 个体评价

在评价阶段,采用基于指数排序的适应度函数[10],每个个体的适应度由它在种群中按目标值排序后的位置确定. 对于种群中的所有个体,按目标值非降或非增的顺序排列. 若个体 i 排在 j 位置,则它的适应度为

$$\mathrm{fit}(i) = \rho(1-\rho)^{j-1},\ j = 1,2,\cdots,\text{pop-size}. \tag{13}$$

其中 $\rho \in (0,1)$,本文取 $\rho = 0.7$.

(4) 交叉操作

在交叉阶段,采用线性次序交叉(LOX)对选中的两个父代个体进行交叉操作. 具体步骤参见文献[9].

(5) 变异操作

本文采用的变异操作方式不同于传统的变异操作,当发生变异时,随机生成新的个体以保持种群的多样性,从而避免种群个体过早收敛.

(6) 保留策略

根据父代种群和子代种群生成下一代种群,并采用$(\mu+\lambda)$策略. 父代种群和子代种群的大小分别为 μ 和 λ,从该$(\mu+\lambda)$的种群中取前 μ 个最佳个体作为下一代父代种群,同时避免重复选择相同的染色体. 这里选择 $\mu=\lambda$.

3.2　算法具体过程

本文设计的两层遗传算法具体过程如下:

Step1:初始化外层种群 $P_X(h)$,$h=1$,对于所有的 $x\in P_X(h)$,令目标值 $q(x)=-\infty$;

Step2:如果 $h\le h_{\max}$,则对所有的 $x\in P_X(h)$ 继续 Step3,否则转到 Step9;

Step3:初始化内层种群 $P_S(k)$,$k=1$;

Step4:如果 $k\le k_{\max}$,则对所有的 $s\in P_S(k)$ 继续 Step5,否则转到 Step7;

Step5:更新 $q(x)=\max\left\{q(x),\ \sum_{i=1}^{n} f_i^s(c_i^x)\right\}$,$g(s)=\sum_{i=1}^{n} f_i^s(c_i^x)$;

Step6:根据 $g(s)$ 进行遗传进化,生成内层的下一代种群 $P_S(k+1)$,$k=k+1$,返回 Step4;

Step7:输出内层最佳解;

Step8:根据 $q(x)$ 进行遗传进化,生成外层的下一代种群 $P_X(h+1)$,$h=h+1$,返回 Step2;

Step9:输出外层最佳解,算法结束.

4　仿真研究

本文针对单机 Just-in-time 系统在加工时间不确定情况下的鲁棒调度问题进行仿真分析. 所生成算例的参数如下:

工件数目 $n\in\{10,20,30,40,50\}$;加工时间的下界为 $\underline{p}_i\in[10,50\theta_1]$ 的均匀分布,$\theta_1\in\{0.2,0.4,0.6,0.8,1.0\}$;加工时间的上界为 $\overline{p}_i\in[\underline{p}_i,(1+\theta_2)\underline{p}_i]$ 的均匀分布,$\theta_2\in\{0.2,0.4,0.6,0.8,1.0\}$;到达率 $\delta\in\{0.25,0.5\}$;到达时间 r_i 为 $[0,\delta E[C_{\max}]]$ 的均匀分布,其中 $E[C_{\max}]=\sum_{i=1}^{n}(\underline{p}_i+\overline{p}_i)/2$;交货期 $d_i=r_i+\gamma p_i$,其中 $\gamma\in[1,5]$ 的均匀分布;提前和拖期权重均为[1,3]的离散均匀分布. 共有 250 种组合方式,每种组合生成 10 个算例,共计 2 500 个算例.

表 1 算法 R 相对于算法 E 的偏移率

算 例	10 jobs		20 jobs		30 jobs		40 jobs		50 jobs	
	AWI	AVI	AWI	AVI	AWI	AVI	AWI	AVI	AWI	AVI
{0.25, *, *}	- 3.51	0.13	- 4.33	0.35	- 2.73	- 0.80	- 2.54	- 0.28	- 1.42	- 0.50
{0.5, *, *}	- 2.78	0.02	- 1.35	0.26	- 4.48	0.32	- 3.68	0.17	- 3.31	0.20
{ *,0.2, *}	- 2.86	0.01	- 1.67	0.24	- 3.01	0.43	- 2.18	0.30	- 1.45	0.18
{ *,0.4, *}	- 3.82	0.20	- 2.16	0.03	- 3.96	- 0.60	- 2.81	0.39	- 1.42	0.36
{ *,0.6, *}	- 3.58	- 0.21	- 2.47	- 0.21	- 3.56	0.39	- 3.32	- 0.61	- 2.03	0.15
{ *,0.8, *}	- 2.21	- 0.02	- 3.07	- 0.35	- 3.47	- 0.50	- 3.10	- 0.40	- 2.41	0.19
{ *,1.0, *}	- 3.55	- 0.20	- 2.86	0.39	- 2.65	0.06	- 3.19	0.11	- 2.61	0.21
{ *, *,0.2}	- 1.20	- 0.40	- 2.08	- 0.57	- 3.23	- 0.92	- 3.67	- 3.02	- 3.51	- 0.74
{ *, *,0.4}	- 2.88	- 0.80	- 2.38	- 0.03	- 3.80	- 1.50	- 3.07	- 1.23	- 2.97	- 0.45
{ *, *,0.6}	- 4.26	0.32	- 3.04	0.13	- 3.55	- 0.40	- 3.20	0.56	- 1.34	0.12
{ *, *,0.8}	- 5.14	0.29	- 2.35	0.27	- 3.47	0.09	- 2.23	0.19	- 1.88	0.07
{ *, *,1.0}	- 1.52	0.25	- 2.92	0.61	- 2.85	0.14	- 3.14	0.08	- 1.98	0.39

本文设计的两层协同进化遗传算法相关参数如下:外层种群和内层种群规模均为40,交叉概率为0.8,变异概率为0.2,停止条件为达到最大进化代数100或连续20次性能没有改善.

利用本文算法和按期望加工时间的确定性调度算法对2 500个算例进行仿真对比,其中对加工时间(这里取其边界值)进行500次模拟实现,以最差性能和平均性能作为比较数据.绝对鲁棒算法R相对于按期望加工时间的确定性调度算法E的最差性能偏移率AWI和平均性能偏移率AVI如表1所示,其中负值表示改善,正值表示恶化.从中可以看出,绝对鲁棒算法在最差性能指标下都优于按期望加工时间的确定性调度算法,对平均性能的恶化程度很小,保证了绝对鲁棒调度仍然具有良好的期望性能.

表1中的数据是仿真结果的统计归纳,统计按照工件数目划分,类型{ *, *, *}表示工件的到达率δ,加工时间下界参数θ_1和加工时间上界参数θ_2.令每个类型下的算例集合为Q,则有

$$\mathrm{AWI} = \frac{\sum_{i\in Q} F_{\mathrm{R}}(i) - \sum_{i\in Q} F_{\mathrm{E}}(i)}{\sum_{i\in Q} F_{\mathrm{E}}(i)} \times 100, \quad (14)$$

$$\mathrm{AVI} = \frac{\sum_{i\in Q} F_{\mathrm{AR}}(i) - \sum_{i\in Q} F_{\mathrm{AE}}(i)}{\sum_{i\in Q} F_{\mathrm{AE}}(i)} \times 100. \quad (15)$$

其中:F_{R}和F_{E}分别为算法R和算法E的最差性能,F_{AR}和F_{AE}分别为算法R和算法E的平均性能.

5 结 语

本文针对单机Just-in-time系统在加工时间不确定的情况下,设计了一种能保证绝对鲁棒性的有效算法.绝对鲁棒性问题实质是一个minimax问题,通过对minimax问题的分析,可知在给定的调度顺序下,在加工时间可行域凸多面体的顶点得到最差性能,从而减小了内层max问题的搜索空间.根据minimax问题的性质,设计出一种两层协同进化遗传算法,外层确定调度顺序,内层在给定调度顺序下搜索得到最差性能的加工时间.对大量算例进行仿真,并与采用期望加工时间的确定性调度算法进行比较,表明本文算法是有效的.

鲁棒调度是处理车间层不确定性问题的有效机制.今后的工作是将该算法推广到更复杂的调度环境,如Flow shop和Job shop.

参考文献(References)

[1] Pinedo M. Scheduling: Theory, algorithms and systems [M]. Englewood Cliffs: Prentice-Hall, 1995.

[2] Daniels R L, Kouvelis P. Robust scheduling to hedge against processing time uncertainty in single-stage production [J]. Management Science, 1995, 41(2): 363-376.

[3] Kouvelis P, Daniels R L, Vairaktarakis G. Robust scheduling of a two-machine flow shop with uncertain processing times [J]. IIE Trans, 2000, 32(5): 421-432.

[4] Yang J, Yu G. On the robust single machine scheduling problem [J]. J of Combinatorial Optimization, 2002, 6(1): 17-33.

[5] 李建更, 涂菶生. 某些调度问题区间摄动鲁棒性研究[J]. 自动化学报, 2001, 27(1): 24-30.
(Li Jian-geng, Tu Peng-sheng. Study on the robustness of some schedules with interval perturbations[J]. Acta Automatica Sinica, 2001, 27(1): 24-30.)

(下转第1159页)

法收敛效果和收敛速度都有较大提高. 本文是根据不同粒子与最优粒子的相似度来改变权重参数,并由此计算粒子群的聚集度,随机初始化粒子的位置,增加粒子群的多样性,提高全局搜索能力. 但本文算法新增加了一些参数,对这些参数如何设置还需进一步讨论.

参考文献(References)

[1] Eberhart R C, Kennedy J. A new optimizer using particle swarm theory[C]. The 6th Int Symp on Micro Machine and Human Science. Nagoya, 1995: 39-43.

[2] Kennedy J, Eberhart R C. Particle swarm optimization [C]. Proc of IEEE Int Conf on Neural Networks. Piscataway: IEEE Service Center, 1995: 1942-1948.

[3] Shi Y, Eberhart R C. A modified particle swarm optimizer [C]. Proc of the IEEE Int Conf on Evolutionary Computation. Piscataway: IEEE Press, 1998: 69-73.

[4] Krink T, Vesterstrom J S, Riget J. Particle swarm optimization with spatial particle extension[C]. Proc of the IEEE Int Conf on Evolutionary Computation. Honolulu: IEEE Inc, 2002: 1474-1497.

[5] Xie X F, Zhang W J, Yang Z L. A dissipative particle swarm optimization[C]. Proc of the IEEE Int Conf on Evolutionary Computation. Honolulu: IEEE Inc, 2002: 1666-1670.

[6] Van Den Bergh. An analysis of particle swarm optimizers [D]. Pretoria: University of Pretoria, 2002: 81 - 83.

[7] Chatterjee A, Siarry P. Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization[J]. Computers and Operations Research, 2006, 33(3): 859-871.

[8] Ratnaweera A, Halgamuge S K, Watson H C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients[J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 240-255.

[9] Zhang L P, Yu H J, Hu S X. Optimal choice of parameters for particle swarm optimization[J]. J of Zhejiang University, 2005, 6(6): 528-534.

[10] He S, Wu Q H, Wen J Y, et al. A particle swarm optimizer with passive congregation[J]. Bio-systems, 2004, 78(1-3): 135-147.

[11] 王凌. 智能优化算法及其应用[M]. 北京: 清华大学出版社, 2001.
(Wang Ling. Intelligent opitmization algorithms with applications[M]. Beijing: Tsinghua University Press, 2001.)

[12] 窦全胜, 周胜光, 马铭. 粒子群优化的两种改进策略[J]. 计算机研究与发展, 2005, 42(5): 897-904.
(Dou Quan-sheng, Zhou Sheng-guang, Ma Ming. Two improvement strategies for particle swarm optimization [J]. J of Computer Research and Development, 2005, 42(5): 897-904.)

(上接第 1154 页)

[6] 李建更, 涂菶生. 一类 Flow shop 调度问题最优调度区间摄动鲁棒性[J]. 控制理论与应用, 2004, 21(1): 25-29.
(Li Jian-geng, Tu Peng-sheng. Interval perturbation robustness of optimal schedules for a class of flow shop problems[J]. Control Theory and Applications, 2004, 21(1): 25-29.)

[7] Kouvelis P, Yu G. Robust discrete optimization and its applications [M]. Dordrecht: Kluwer Academic Publishers, 1997.

[8] Lenstra J, Rinnooy K A, Brucker P. Complexity of machine scheduling problems[J]. Annals of Discrete Mathematics, 1977, 1: 343-362.

[9] 玄光男, 程润伟. 遗传算法与工程优化 [M]. 于歆杰, 等译. 北京: 清华大学出版社, 2004.
(Gen Mitsuo, Cheng Run-wei. Genetic algorithm and engineering optimization [M]. Yu Xin-jie, et al (translators). Beijing: Tsinghua University Press, 2004.)

[10] Michalewicz Z. Genetic algorithm + data structure = evolution programs[M]. New York: Springer-Verlag, 1994.