

Flow shop问题的嵌套分区优化调度方法

武 维, 管晓宏, 卫军胡

(西安交通大学 制造系统工程国家重点实验室, 系统工程研究所, 陕西 西安 710049)

摘要: 介绍了嵌套分区算法(NP)的基本思想, 并用于求解流水作业优化调度问题. 算法用嵌套分区树来描述流水作业调度问题, 对可行域进行系统性分区, 然后集中搜索有优良解的区域. 在每一步迭代中, 算法跟踪最有希望的分区, 并结合启发式算法和邻域搜索来实现分区转移. 仿真实验表明, 该算法比单纯的启发式算法和邻域搜索有较好的寻优能力.

关键词: 嵌套分区算法; 流水作业调度; 优化调度; 启发式算法

中图分类号: TP391 **文献标识码:** A

Nested partition-optimization algorithm for flow shop scheduling

WU Wei, GUAN Xiao-hong, WEI Jun-hu

(State Key Laboratory for Manufacturing Systems Engineering, Systems Engineering Institute,
Xi'an Jiaotong University, Xi'an Shaanxi 710049, China)

Abstract: A nested partition-algorithm and its application to the flow shop scheduling problem are addressed in this paper. In this application, the flow shop scheduling problem is considered as a partition tree. The algorithm performs an iterative process in which it partitions the feasible region and searches for the subset where the optimal solution is located. In the iteration, a heuristic algorithm and a neighborhood-search are employed to select a sample-point for estimating the promising index for each subset. Finally, computational experiments indicate that the algorithm outperforms the constructive heuristic algorithm and neighborhood-search in the performance of searching.

Key words: nested partition-algorithm; flow shop scheduling; optimization scheduling; heuristic algorithm

1 引言(Introduction)

生产调度是制造系统的一个研究热点, 也是理论研究中最为困难的问题之一. 优良的调度策略对于提高生产系统的最优性、提高经济效益, 有着极大的作用. 目前, 对于调度理论的研究已受到广泛的关注, 并取得了较大的进展, 但还很不成熟. 多数调度问题都属于NP难题, 目标解的搜索涉及解空间的组合爆炸. 线性规划、动态规划、分支定界和梯度下降等传统方法, 要么需要目标函数的特殊信息, 要么复杂度大, 优化性能差, 因而一般只能处理小规模问题, 难以高效地求解复杂问题. 针对基于计算和数值式优化技术的弱点, 研究和开发了统计式全局搜索和人工智能的方法, 例如模拟退火、遗传算法、禁忌搜索、神经网络等, 较满意地解决了一大批传统优化方法难以解决的复杂问题. Shi等学者提出的嵌套分区(nested partitions, NP)算法结合了全局优化和局部

优化的特点, 具有开放性、全局性、并行性和易操作性等突出优点, 能够解决许多确定性和随机性的仿真优化问题, 具有很高的计算效率^[1]. 他们把NP算法应用于TSP问题、供应链管理、机器分配、产品设计、客户关系管理等领域, 取得了显著效果^[2]. 国内关于嵌套分区算法的文献还比较少, 本文用嵌套分区算法求解flow shop的调度问题, 最后通过实验证明算法的计算效率和可行性^[3].

2 Flow shop 调度问题描述(Flow shop scheduling problem description)

Flow shop调度问题(flow shop scheduling problem, FSP)是许多实际流水线生产调度问题的简化模型, 也是一个典型的NP-hard问题. FSP可以描述为: m 台机器对 n 个工件的流水加工过程, 每个工件在各机器上加工顺序相同, 同时约定, 每个工件在每台机器上只加工一次, 每台机器在某一时刻只能

加工一个工件,各工件在各机器上所需的加工时间已定,要求得到一个调度方案使得某项性能指标最优^[4].若约定每台机器对各工件的加工顺序相同,则称其为置换flow shop调度问题.本文研究嵌套分区算法求解置换flow shop调度问题^[5].

置换flow shop调度问题的数学描述如下.令 t_{ij} 为工件 i 在机器 j 上的加工时间. T_{ij} 为工件 i 在机器 j 上的加工完成时刻.不失一般性,假设各工件按机器1至 m 的顺序进行加工,令 $\pi = (\sigma_1, \sigma_2, \dots, \sigma_n)$ 为所有工件的一个排序,以上定义和假设用如下的数学模型表示^[6]:

$$\pi^* = \arg \{f(\pi) = T_{\sigma_n, m}\} \rightarrow \min, \quad (1)$$

s.t.

$$T_{\sigma_1, 1} = t_{\sigma_1, 1}, \quad (2)$$

$$T_{\sigma_j, 1} = T_{\sigma_{j-1}, 1} + t_{\sigma_j, 1}, j = 2, \dots, n, \quad (3)$$

$$T_{\sigma_1, i} = T_{\sigma_1, i-1} + t_{\sigma_1, i}, i = 2, \dots, m, \quad (4)$$

$$T_{\sigma_j, i} = \max \{T_{\sigma_{j-1}, i}, T_{\sigma_j, i-1}\} + t_{\sigma_j, i}. \quad (5)$$

式(1)表示以最大完成时间为指标,即Makespan指标,为作业 σ_n 在机器 m 上加工完成的时刻.式(2)表示作业 σ_1 在机器1上加工完成的时刻.式(3)表示作业 σ_j 在机器1上加工完成时刻.式(4)表示作业 σ_1 的加工顺序约束,作业 σ_1 的第 i 道工序必须在第 $i-1$ 道工序完成后才能开始.式(5)表示作业 σ_{j-1} 在机器 i 加工完成后和作业 σ_j 在机器 $i-1$ 加工完成后,才可以在机器 i 上加工作业 σ_j ^[7,8].

3 嵌套分区算法(Nested partitions algorithm)

DEDS的优化问题通常可以描述为如下形式:

$$\min_{\theta \in \Theta} J(\theta).$$

其中: Θ 是可行域, $J: \Theta \rightarrow \mathbb{R}$ 是系统的性能函数.假设该DEDS有唯一最优解 $\theta_{\text{opt}} \in \Theta$,对所有的 $\theta \in \Theta \setminus \{\theta_{\text{opt}}\}$ 有 $J(\theta_{\text{opt}}) < J(\theta)$.在复杂DEDS中, $J(\theta)$ 通常是对输入参数的性能估计.但在有些情况下没有解析表达式用以表达系统的性能函数.这样用仿真的方法, $J(\theta) = E[L(\theta)]$ 来估计系统的性能.其中 $L(\theta)$ 是对系统性能的无偏估计,即系统抽样的性能.NP算法对系统性能函数的形式没有要求.为了说明嵌套分区算法,给出如下定义:

单解域 如果一个集合中所有解都是可行的,则该集合是一个可行域(valid region).所有可行域的集合表示为 Σ .只包含一个可行解的区域称为单解域(singleton region).单解域的集合表示为 Σ_0 , $\Sigma_0 \subset \Sigma$.

裙域 算法在迭代过程中,对 $\sigma(k)$ 进行分区的同时亦将 $\sigma(k)$ 外的其他分区合并为一个区域 $\Theta \setminus \sigma(k)$,称为裙域(surrounding region).

母域 对可行域 $\eta \in \Sigma$ 分区,得到可行域 $\sigma \in \Sigma$,则称 σ 为 η 的子域(subregion), η 为 σ 的母域(super region).定义母域映射函数 $s: \Sigma \rightarrow \Sigma$,即如果 $\sigma \in \Sigma \setminus \{\Theta\}$,则 $s(\sigma) = \eta \in \Sigma$.对于 Θ 定义 $s(\Theta) = \Theta$.

深度 定义 $d: \Sigma \rightarrow N_0$ 为区域 Σ 的分区深度(depth),即由区域 Θ 到区域 Σ 的分区层数.例如, Θ 的深度为0,即 $d(\Theta) = 0$.则其子域深度为1,依此类推.单解域具有最大深度.

希望指数 设定希望指数(promising index) $I: \Sigma \rightarrow \mathbb{R}$,用来选择包含 θ_{opt} 的区域,以确定下一步被分区的区域.希望指数要求对 $\sigma = \{\theta\} \in \Sigma_0$ 有 $I(\sigma) = J(\theta)$.NP算法具有开放性,可以根据具体问题设定希望指数的形式.

NP算法的基本思想是,在算法的 k 次迭代中,如果认为 $\sigma(k) \subseteq \Theta$ 是包含 θ_{opt} 的可行域,则把 $\sigma(k)$ 分区为 $M_{\sigma(k)}$ 个子域.并把 $\Theta \setminus \sigma(k)$ 定义为裙域.这样就得到 $M_{\sigma(k)} + 1$ 个不相交的子域(初次分区除外).在每个区域上利用抽样方法估计该区域的性能,确定最优解可能出现的区域.然后对该区域进行分区,直到得到单解域^[9,10].

NP算法分为4个步骤:分区(partition)、抽样(sampling)、选区(selection)、回溯(backtrack).

分区 如果 $\sigma(k) \notin \Sigma_0$,则分 $\sigma(k)$ 为 $M_{\sigma(k)}$ 个子域 $\sigma_1(k), \dots, \sigma_{M_{\sigma(k)}}(k)$.并把 $\Theta \setminus \sigma(k)$ 合为一个区域 $\sigma_{M_{\sigma(k)}+1}(k)$,即产生 $M_{\sigma(k)} + 1$ 个区域.但是如果 $\sigma(k) = \Theta$,则只分区产生 $M_{\sigma(k)}$ 个区域.

抽样 对区域 $\sigma_j(k)$ 选择 $N_{\sigma_j(k)}$ 个抽样点 $\theta^{j1}, \theta^{j2}, \dots, (\theta^{jN_{\sigma_j(k)}})$,并估计其性能值为 $L(\theta^{j1}), L(\theta^{j2}), \dots, L(\theta^{jN_{\sigma_j(k)}})$.算法可采用多种抽样方法,只要满足区域中的每个点被选择的概率大于0.

选区 使用希望指数 $I: \Sigma \rightarrow \mathbb{R}$ 来确定下一步要分区的区域.并用如下公式估计性能参数:

$$I(\eta) = \lim_{\theta \in \eta} J(\theta).$$

回溯 如果区域 $\sigma_j(k) \subset \Theta \setminus \sigma(k)$ 是要被分区的区域,则算法将要回溯.有两种回溯策略:第1种为回溯到当前域的母域,即 $\sigma(k+1) = s(\sigma(k))$;第2种为回溯到初始 Θ 域,即 $\sigma(k+1) = \Theta$.

4 调度流程算法描述(Scheduling algorithm description)

求解flow shop调度问题的整体流程如图1所示.

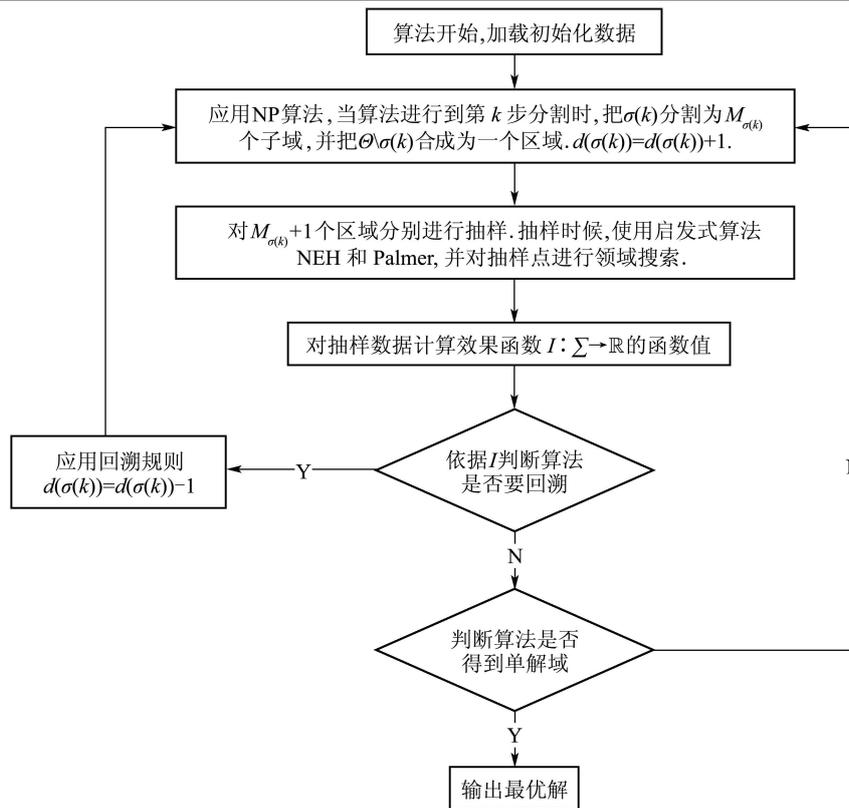


图 1 嵌套分区调度算法流程图

Fig. 1 Nested partitions scheduling algorithm flow chart

4.1 调度分区(Scheduling partition region)

嵌套分区算法求解flow shop调度问题的过程,就是构建分区树的过程,在分区树每一层调度完成一个作业.这样在深度为d的分区上有d个作业已经被调度, n - d个作业没有被调度.一个完整的调度结果可以表示为

$$\theta = (J) = \{j_1, j_2, \dots, j_n\}.$$

图2说明了作业数n = 3的分区树构建过程.

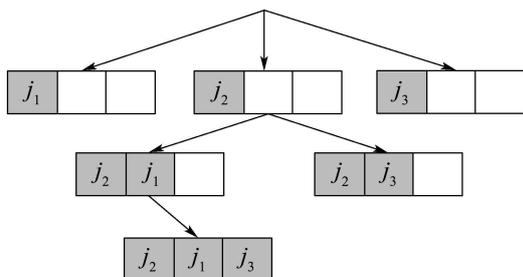


图 2 作业数n = 3的分区树构建过程

Fig. 2 Part of partition tree with job n = 3

分区算子在分区树的每个深度d调度完成一个作业.这样当作业数为n时, d = n. 深度为0的子域定义为

$$\{(j_1), (\cdot), \dots, (\cdot)\},$$

$$\{(j_2), (\cdot), \dots, (\cdot)\},$$

⋮

$$\{(j_n), (\cdot), \dots, (\cdot)\}.$$

深度为d的子域与深度为0的子域表示方式类似,区别在于深度为d分区已经固定了d个作业.分区树上子域数量随着深度的增加数量逐渐减少,其数学表达式为 $|N_j^U|$.

4.2 调度抽样算子(Scheduling sampling operator)

Flow shop调度抽样算子用来确定下一步被调度的作业.一个有效的抽样算子,要保证上步分区中的每个作业都有被选中的概率.如果在区域σ中抽样,因为已有d(σ)个作业被调度,这样需要抽样剩下的n - d(σ)个作业,并生成抽样点:

$$\theta = \{j_1, j_2, \dots, j_n\}.$$

本文使用启发式算法对区域σ进行数据抽样.分别研究了Palmer算法和NEH算法在嵌套分区算法中的抽样过程.并对两算法产生的抽样点进行邻域搜索.

Palmer算法抽样 记 t_{ij} 为工件i在机器j上的加工时刻,该方法对每个工件计算参量S(i):

$$S(i) = \sum_{j=1}^m (2j - m - 1)t_{ij}/2,$$

$$i = 1, \dots, n - d(\sigma), j = 1, \dots, m.$$

将工件参量 $S(i)$ 值按递减的顺序进行排列,从而得到区域 σ 中的一个调度抽样点.

NEH算法抽样 该算法基本思想是一个工件的总加工时间越长,它应具有越高的优先权.算法的具体描述如下^[11]:

- 1) 对区域 σ 中的每个工件 i 计算它的总加工时间 $T_i, T_i = \sum_{j=1}^m t_{ij}$;
- 2) 对工件按 T_i 递减的顺序排列,得到序列 P ;
- 3) 选出在 P 序列中位于第1、第2位的两个工件,分别计算这两个工件两种排列下的最大加工时间 F_{\max} ,取其 F_{\max} 小的排列,将这两个工件的相对位置固定下来,放入序列 Q 中.令 $k = 3$;
- 4) 从 P 序列中选出第 k 位置的工件,将其插入到 Q 序列中的 k 个位置中,找出 F_{\max} 最小的排列,放入序列 Q 中;
- 5) 若 $k = n$,转步骤6),否则,令 $k = k + 1$,转步骤④;
- 6) 序列 Q 即为调度抽样点.

邻域搜索 对Palmer算法和NEH算法产生的调度抽样点进行邻域搜索^[12].

设 $\theta = (J) = \{j_1, j_2, \dots, j_n\}$ 是区域 σ 中的一个抽样点.交换分区 σ 中的两个作业 θ_{p_1} 和 θ_{p_2} ,并且保证 $d(\sigma) \leq d(\theta_{p_1}) < d(\theta_{p_2}) \leq n$.交换策略如图3所示,图中swap表示交换.

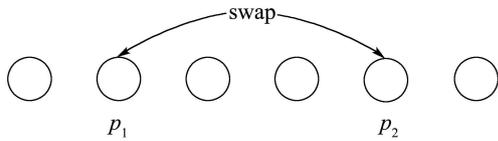


图3 邻域搜索策略
Fig. 3 Neighborhood search strategy

域搜索采用“见好就收”的策略.算法以某种顺序搜索当前点的邻域,碰到比当前点好的点,就走过去,直到所有邻域遍历完毕.

4.3 调度选区算子(Scheduling selection region operator)

使用希望指数 $I: \Sigma \rightarrow \mathbb{R}$ 来评定下一步要选择的分区.希望指数要和单解域的目标函数一致.这里采用目标函数

$$\pi^* = \arg \{f(\pi) = T_{\sigma_n, m}\} \rightarrow \min$$

作为希望指数.选择性能指标最小的分区作为下一步要分区的区域.

4.4 调度回溯策略(Scheduling backing strategy)

当算法经过 k 次分区后,如果当前分区 $\sigma(k)$ 以外的整个区域的希望指数是最小的,则回溯到

$\sigma(k)$ 的母域 $s(\sigma(k))$.对 $s(\sigma(k))$ 进行分区、抽样和选区的操作.具体策略为:保存已经调度作业的路径.回溯时,将已调度的作业数减1,深度减1即可.

5 仿真试验及结果分析(Simulation and analysis)

为了说明嵌套分区算法对置换flow shop调度问题的寻优能力,本文设计了两组实验,分别在Car类和Rec类标准测试问题集上进行验证.两组实验的区别在于抽样过程中采用了不同的启发式算法.仿真程序使用VC++6.0编写,以Inter Celeron 1.6GHz计算机为运行环境,进行实验.两组实验见表1和表2.

表1 实验1
Table 1 Experiment 1

| 版本 | 算法名 | 描述 |
|----|--------|-------------------|
| A | Palmer | 即Palmer原始算法 |
| B | PNP1 | 使用Palmer算法抽样 |
| C | PNP2 | 使用Palmer算法和邻域搜索抽样 |

表2 实验2
Table 2 Experiment 2

| 版本 | 算法名 | 描述 |
|----|------|----------------|
| D | NEH | 即NEH原始算法 |
| E | NNP1 | 使用NEH算法抽样 |
| F | NNP2 | 使用NEH算法和邻域搜索抽样 |

表1是对实验1算法的分类,表2是对实验2算法的分类.对实验1和实验2分别在Car类和Rec类标准测试问题集上进行仿真实验.表3是实验1的实验结果,表4是实验2的实验结果.在表3和表4中 N 表示job数量; M 表示machine数量; C^* 为该问题的最优解; t 为计算时间,单位为s; RE 表示实验结果与 C^* 的相对误差 $(C^{np} - C^*)/C^*$.

图4是实验1中PNP2算法和实验2中NNP2算法的结果比较.

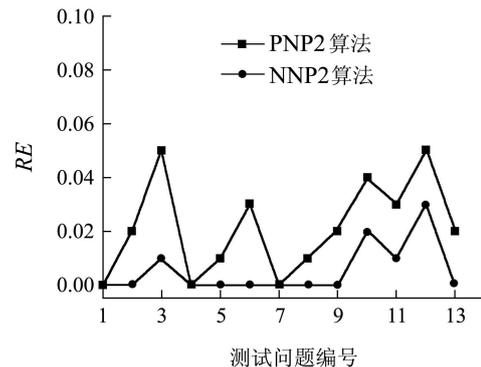


图4 PNP2和NNP2算法结果比较
Fig. 4 PNP2 and NNP2 comparative results

表3 实验1结果

Table 3 Results of experiment 1

| 问题 | N | M | C* | Palmer | PNP1 | | PNP2 | |
|-------|----|----|------|--------|------|---|------|------|
| | | | | RE | RE | t | RE | t |
| Car1 | 11 | 5 | 7038 | 0 | 0 | 0 | 0 | 0 |
| Car2 | 13 | 4 | 7116 | 0.13 | 0.12 | 0 | 0.02 | 0 |
| Car3 | 12 | 5 | 7312 | 0.09 | 0.08 | 0 | 0.05 | 0 |
| Car4 | 14 | 4 | 8003 | 0.05 | 0.01 | 0 | 0 | 0 |
| Car5 | 10 | 6 | 7720 | 0.05 | 0.04 | 0 | 0.01 | 0 |
| Car6 | 8 | 9 | 8505 | 0.13 | 0.11 | 0 | 0.03 | 0 |
| Car7 | 7 | 7 | 6590 | 0.15 | 0 | 0 | 0 | 0 |
| Car8 | 8 | 8 | 8366 | 0.06 | 0.04 | 0 | 0.01 | 0 |
| Rec03 | 20 | 5 | 1109 | 0.08 | 0.05 | 0 | 0.02 | 0.2 |
| Rec09 | 20 | 10 | 1537 | 0.22 | 0.15 | 0 | 0.04 | 0.3 |
| Rec15 | 20 | 15 | 1950 | 0.11 | 0.06 | 0 | 0.03 | 0.4 |
| Rec27 | 30 | 15 | 2373 | 0.11 | 0.08 | 0 | 0.05 | 3.1 |
| Rec35 | 50 | 10 | 3227 | 0.07 | 0.05 | 0 | 0.02 | 25.2 |

表4 实验2结果

Table 4 Results of experiment 2

| 问题 | N | M | C* | NEH | PNP1 | | PNP2 | |
|-------|----|----|------|------|------|------|------|------|
| | | | | RE | RE | t | RE | t |
| Car1 | 11 | 5 | 7038 | 0 | 0 | 0 | 0 | 0 |
| Car2 | 13 | 4 | 7116 | 0.03 | 0 | 0 | 0 | 0 |
| Car3 | 12 | 5 | 7312 | 0.02 | 0.01 | 0 | 0.01 | 0 |
| Car4 | 14 | 4 | 8003 | 0 | 0 | 0 | 0 | 0 |
| Car5 | 10 | 6 | 7720 | 0.04 | 0.01 | 0 | 0 | 0 |
| Car6 | 8 | 9 | 8505 | 0.04 | 0 | 0 | 0 | 0 |
| Car7 | 7 | 7 | 6590 | 0.06 | 0.01 | 0 | 0 | 0 |
| Car8 | 8 | 8 | 8366 | 0.01 | 0.01 | 0 | 0 | 0 |
| Rec03 | 20 | 5 | 1109 | 0.07 | 0 | 0.1 | 0 | 0.3 |
| Rec09 | 20 | 10 | 1537 | 0.07 | 0.04 | 0.2 | 0.02 | 0.5 |
| Rec15 | 20 | 15 | 1950 | 0.05 | 0.03 | 0.2 | 0.01 | 0.6 |
| Rec27 | 30 | 15 | 2373 | 0.09 | 0.05 | 1.5 | 0.03 | 4.6 |
| Rec35 | 50 | 10 | 3227 | 0.05 | 0 | 12.6 | 0 | 38.2 |

从表3中可以看到, PNP1算法比原始Palmer算法有更好的寻优能力, 而PNP2算法中加入邻域搜索后比PNP1算法有更好的寻优能力. 从表4中可以看到, NNP1算法比原始NEH算法有更好的寻优能力, 而NNP2算法中加入邻域搜索后比PNP1算法有更好的寻优能力. 可见在嵌套分区算法中加入启发式算法和邻域搜索后大大的加强了算法的寻优能力. 从图4中可以看到NNP2算法的实验结果明显优于PNP2的实验结果. 由于NEH算法的实验结果优于Palmer算法, 可见在嵌套分区算法中加入更加有效的启发式算法NEH后, 嵌套分区算法有更好的寻优能力.

6 结语(Conclusion)

本文提出在嵌套分区算法(NP)框架下, 结合启

发式算法和邻域搜索算法, 解决极小化总完工时间置换流水作业调度问题的新型算法. 仿真结果表明其寻优能力大大优于单纯的启发式算法和邻域搜索算法. 可见嵌套分区算法在求解组合优化问题时较好的寻优能力.

参考文献(References):

- [1] SHI L, 'OLAFSSON S. Nested partitions method global optimization[J]. *Operations Research*, 2000, 48(3): 309 – 407.
- [2] SHI L, 'OLAFSSON S, CHEN Q. An optimization framework for product design[J]. *Management Science*, 2001, 47(2): 1681 – 1692.
- [3] 'OLAFSSON S, SHI L. A method for scheduling in parallel manufacturing systems with flexible resources[J]. *IIE Transactions Research*, 2000, 32(1): 135 – 146.
- [4] TAILLARD E D. Some efficient heuristic methods for the flow shop sequencing problem[J]. *European Journal of Operation Research*, 1990, 47(1): 65 – 74.
- [5] GUPTA J N D. A functional heuristic algorithm for flowshop scheduling problem[J]. *Operational Research Quarterly*, 1971, 22(1): 39 – 47.
- [6] WANG L, ZHENG D Z. A modified evolutionary programming for flow shop scheduling[J]. *International Journal of Advanced Manufacturing Technology*, 2003, 22(7/8): 522 – 527.
- [7] SHI G Y. A genetic algorithm applied to a class job shop scheduling problem[J]. *International Journal of System Science*, 1997, 28(1): 25 – 32.
- [8] 李建更, 涂莘生. 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 Jiangeng, TU Fengsheng. Interval perturbation robustness of optimal schedules for a class of Flow Shop problems[J]. *Control Theory & Applications*, 2004, 21(1): 25 – 29.)
- [9] SHI L, 'OLAFSSON S, CHEN Q. A new hybrid optimization algorithm[J]. *Computer & Industrial Engineering*, 1999, 36(1): 409 – 426.
- [10] 'OLAFSSON S, SHI L. Ordinal comparison via the nested partitions method[J]. *Discrete Event Dynamic Systems*, 2002, 12(1): 211 – 239.
- [11] NAWAZ M, ENSCORE E Jr, HAM I. A heuristic algorithm for the m-machine, n-job flowshop sequencing problem[J]. *International Journal of Management Science*, 1983, 11(1): 91 – 95.
- [12] WANG C, CHU C, PROTH J M. Heuristic approaches for scheduling problems[J]. *European Journal of Operational Research*, 1997, 96(3): 636 – 644.

作者简介:

武维 (1982—), 男, 博士研究生, 主要研究方向为复杂系统的建模、仿真与优化调度, E-mail: mr.allawn@gmail.com;

管晓宏 (1955—), 男, 教授, 博士生导师, 主要研究方向为系统优化调度、计算机网络信息安全, E-mail: xhguan@sei.xjtu.edu.cn;

卫军胡 (1966—), 男, 副教授, 主要研究方向为复杂系统建模与仿真、分布式仿真, E-mail: jhwei@sei.xjtu.edu.cn.