

# 调度问题微结构及柔性优化方法<sup>1)</sup>

沈益民<sup>1,2</sup> 范玉顺<sup>1</sup>

<sup>1</sup>(清华大学自动化系 北京 100084)

<sup>2</sup>(成都电子机械高等专科学校 成都 610015)

(E-mail: chen-ym04@mails.tsinghua.edu.cn)

**摘要** 多处理器、多模式等调度问题的出现, 标志着调度问题正呈现出更加复杂多样、更强调柔性的发展趋势. 本文采用嵌套的微结构方法, 对现有调度问题进行统一表述, 并且能够表述一些传统方法无法表述的复杂问题. 针对主从微结构, 本文用总柔性度 (GFD) 作为调度问题柔性的衡量指标, 并按照 GFD 的分类给出了求解调度问题的线性规划方法. 定量实验不仅表明适当的柔性能有效优化调度问题的最优解, 而且针对车间设备群规模合理性给出了统计结论.

**关键词** 调度, 柔性, 微结构, 主从式

**中图分类号** TP202.7

## Microstructure and Flexible Optimizing Method of Scheduling Problem

SHEN Yi-Min<sup>1,2</sup> FAN Yu-Shun<sup>1</sup>

<sup>1</sup>(Automation Department, Tsinghua University, Beijing 100084)

<sup>2</sup>(Chengdu Electromechanical College, Chengdu 610015)

(E-mail: chen-ym04@mails.tsinghua.edu.cn)

**Abstract** The appearance of multi-processor and multi-mode scheduling problems marks a development trend that becomes more complicated and flexibility-oriented. Through recursive microstructure method, the thesis offers a uniform definition for the current scheduling problem as well as some complex problems that are hard to be clarified with traditional methods. In terms of principal-subordinate structure, general flexibility degree (GFD) is used here as an indicator to measure the scheduling flexibility, and GFD classification provides a ground for linear and integer programming solution to the problem. Quantitative experiment not only verifies that appropriate flexibility can effectively optimize the best solution of the scheduling problem, but also results in a statistical conclusion towards the reasonable machine group size.

**Key words** Scheduling, flexibility, microstructure, principal-subordinate

## 1 问题的提出

调度问题的目标是寻求需求 (任务) 和资源 (机器设备等) 在某一优化目标下的最佳分配关系, 而在各种资源中, 设备总是处于核心的位置. 伴随着单一品种、大批量生产向多品种、小批量转化的趋势, 在敏捷制造、协同工作的环境下, 设备调度的“柔性”受到了越来越多的关注:

1) 国家“863”计划项目 (2005AA411910) 资助

Supported by National “863” Project (2005AA411910)

收稿日期 2005-3-31 收修改稿日期 2005-11-12

Received March 31, 2005; in revised form November 12, 2005

- 1) 多处理器、多模式调度问题成为了新的研究重点<sup>[1~5]</sup>;
- 2) 围绕调度柔性化的计算技术不断涌现<sup>[6~9]</sup>;
- 3) 网格计算、无线传感器网络等新技术为柔性的车间调度提供了技术手段.

但以下两个问题没有得到解决:

- 1) 一些更为复杂的问题, 例如主从结构问题, 无法在现有框架下处理. 更进一步说, 如何统一地描述和解决正在涌现和将要涌现的新的调度问题, 缺乏一致的方法.
- 2) 缺乏对于设备柔性的准确定义, 亦缺乏柔性程度的量化以及其对调度优化结果影响的定量研究.

本文定义了调度问题的微结构, 使用微结构不仅可以对大部分调度问题给出统一的表述, 而且能够表述主从结构等复杂设备环境下的调度问题. 基于微结构, 本文定义了衡量设备柔性的指标  $GFD$ , 并针对低柔性、高柔性两种情况给出了求解问题的线性规划方法. 本文最后还采用实验数据验证了采用柔性结构对于调度问题的改进效果.

## 2 调度问题的微结构表示

对于一个多设备调度任务实例, 有  $n$  项独立任务需要加工:  $T = \{T_1, T_2, \dots, T_n\} = \{T_j\}_{j=1}^n$ . 其可用设备集  $M = \{M_i = (p_{i1}, p_{i2}, \dots, p_{in})\}_{i=1}^m$  是  $n$  维空间  $R^n$  上的离散点集, 这里  $p_{ij}$  表示设备  $M_i$  加工任务  $T_j$  的时间,  $p_{ij} = \infty$  当且仅当  $T_j$  任务不能在  $M_i$  设备上加工.

**定义 1.** 设  $(F, ar)$  是型, 使  $R^n$  构成  $F$  型-泛代数. 若  $\exists R^n$  上有界点集族  $\{U_i\}_{i=1}^m$ , 对  $\forall i \neq j$ , 有  $U_i \cap U_j = \phi$ , 且  $\exists \{f_i\}_{i=1}^m \subset F$ ,  $ar(f_i) = |U_i|$ , 使  $f_i(U_i) = M_i$ , 则称  $U_i$  是  $M_i$  在  $f_i$  下的微结构 (microstructure, MS),  $\{U_i\}_{i=1}^m$  是  $M$  在  $\{f_i\}_{i=1}^m$  下的微结构.

为避免混淆, 称  $U_i$  的元素  $u_i$  为设备, 而称  $M_i$  为设备群. 事实上, 由于微结构具有可嵌套性, 因此在不同的层次上, 设备和设备群是可以相互转化的.

**定义 2.** 若  $M$  具有两种不同的微结构  $\{U_i\}, \{V_i\}$ , 且  $\bigcup_{i=1}^m U_i = \bigcup_{i=1}^m V_i$ , 则称  $M$  具有柔性微结构 (Flexible MS, FMS), 或简称柔性的. 相反, 则称为具有刚性微结构 (Rigid MS, RMS), 或简称刚性的.

特别的, 定义  $R^n$  上的二元算子  $\cup, \cap \in T$ , 对于  $u = (p_1, p_2, \dots, p_n), v = (q_1, q_2, \dots, q_n) \in R^n$ ,  $u \cup v = (\frac{1}{1/p_j + 1/q_j})_{j=1}^n = (\frac{p_j q_j}{p_j + q_j})_{j=1}^n$ ,  $u \cap v = (\max(p_j, q_j))_{j=1}^n$ , 分别称  $\cup, \cap$  为并联、串联算子. 有限个并、串联算子在复合运算  $\circ$  下构成串并混合算子, 简称混合算子.

调度问题中的设备群, 在不同的算子下, 构成不同的微结构. 图 1 中, 设备群  $M_{i-1}$  具有串联微结构,  $M_{i+1}$  具有并联微结构,  $M_i$  具有主从式混合微结构.

按照  $\alpha|\beta|\gamma$  分类法将调度问题中部分具代表性的问题, 列出其微结构如表 1.

表 1 调度问题微结构分类  
Table 1 Classifying scheduling problems according to microstructure

部分具代表性的调度问题		$\alpha \beta \gamma$ 示例	微结构
单一工序问题	同等处理机	$\alpha = P_m$	平凡微结构
	并行处理机	$\alpha = Q_m$	平凡微结构
	相异处理机	$\alpha = R_m$	平凡微结构
多工序问题	开放车间	$\alpha = O$	刚性串联微结构
	流程车间	$\alpha = F$	刚性串联微结构
	任务车间	$\alpha = J$	柔性串联微结构
多处理机问题	确定多处理机	$\beta = fix_j$	刚性并联微结构
	并行多处理机	$\beta = size_j$	柔性并联微结构
	多模式多处理机	$\beta = set_j$	柔性并联微结构
复杂处理机问题	-	-	刚性混合微结构
	-	-	柔性混合微结构

可以看出, 仅仅采用两种特殊的算子, 就能表达大部分的调度问题, 甚至能够表达  $\alpha|\beta|\gamma$  分类法不能覆盖的复杂处理机问题. 以下, 重点讨论较为复杂的主从混合微结构问题, 优化目标函数取  $C_{\max}$ .

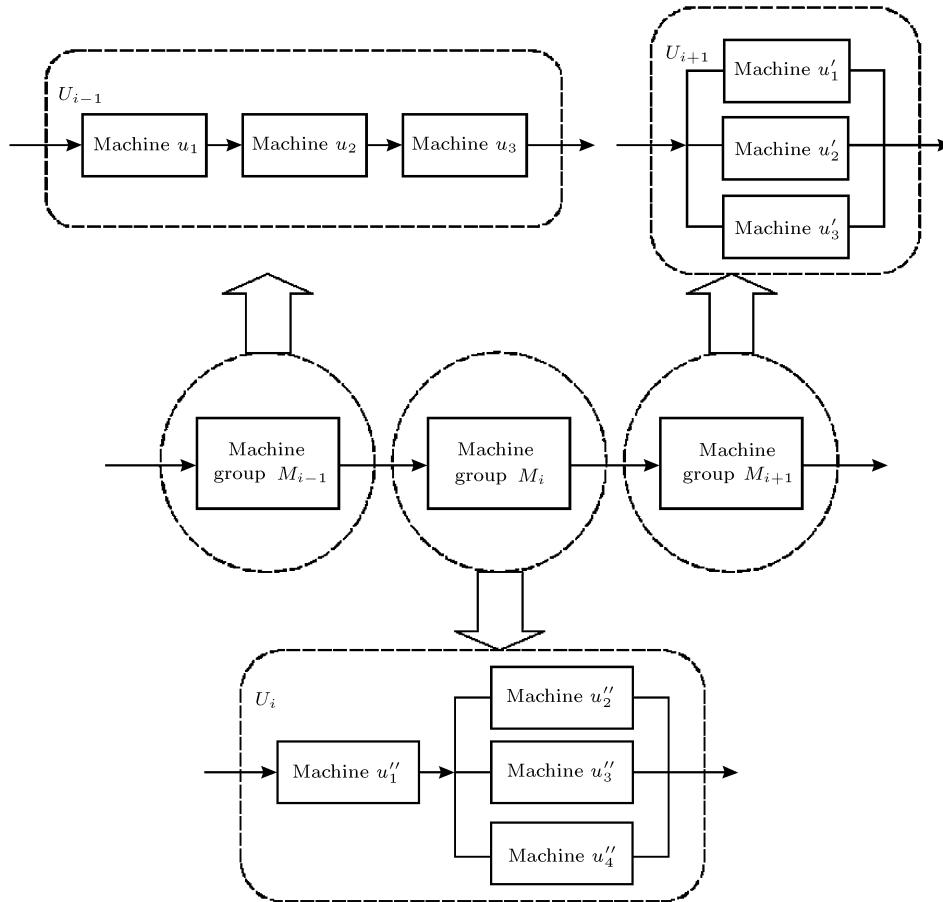


图 1 串联、并联、混合微结构

Fig. 1 Series-wound, shunt-wound and hybrid microstructure

### 3 主从微结构及其柔性

**定义 3.**  $\cap \circ (\cup \circ \cup \circ \dots \cup)$  称为主从式算子, 调度问题对应混合主从式算子中串联算子左元的设备称为主机, 其他设备称为从机.

主从式算子构成的主从微结构 (Principal-subordinate microstructure, PSM) 问题, 对应于实际调度环境中的以下情况:

- 1) 由具有一定传输能力约束的传输设备连接加工设备, 传输设备与加工设备具有非对称性;
- 2) 由一台主机所管控的多台服务器, 主机和服务器之间具有管理、服务层次的差异;
- 3) 在作业中所使用的某种关键资源, 比如设计小组中的主设计师、具有特殊加工能力的机床、连接主要消耗原料的管线等, 均不能等同于组内的其他资源.

带有资源约束的多处理机问题<sup>[10]</sup>与 PSM 问题有较大的相似之处, 但前者无法表出后者, 原因在于:

- 1) 资源的可获得性对于所有的设备是一致的, 而 PSM 问题中的一台主机与不同从机

之间存在距离、连接的差异, 往往只能被几台从机所获得. 这样的限制在目前的资源约束问题中很难描述, 更无人解决.

2) 资源没有“速度”, 因此不会成为瓶颈. 而主机在组内承担特定的管理、调度、传输等任务, 因此其速度往往会制约设备组的运行.

**定义 4.** 从机  $u_l$  可属的设备群的数量为从机柔性度 (flexibility), 并记为  $f_l$ . 所有从机的从机柔性度乘积为总柔性度 (General flexibility degree,  $GFD$ ), 即:

$$GFD = \prod_l f_l \tag{1}$$

在不同的微结构下, 同一从机可以属于不同的设备群. 由于总可以假设每一从机至少属于某一设备群, 因此  $GFD$  是一个正整数.

$GFD$  是 PSM 问题的柔性度量. 当  $GFD=1$  时, 由 (1) 式可知对  $\forall l$  有  $f_l = 1$ , 即每一从机只能属于同一设备群, 因此这是一种刚性的结构. 刚性微结构下, 算子建立起了设备及其微结构之间的准确关系, 调度问题可以逐层求解.

随着  $GFD$  的增长, 柔性则不断上升, 问题的难度也增强. 以下分别对柔性较小和柔性较大两种情况进行讨论.

## 4 柔性主从微结构调度问题的求解

### 4.1 低柔性主从微结构

$GFD = 2$  时, 不失一般性, 设从机  $u_1$  是唯一的双柔性度从机, 其可属的两设备群为  $M_1$  和  $M_2$ . 由于其他设备群及其从机均为刚性结构, 执行任务和  $M_1, M_2$  相互分离, 因此对于任何一种调度方案, 总可以对  $M_1$  和  $M_2$  承担的任务的先后顺序进行调整, 使存在一个时刻  $t$ , 在  $t$  时刻之前,  $u_1$  一直和  $M_1$  连接, 而  $t$  时刻之后, 则改和  $M_2$  连接. 从而, 只需要在包含一次设备切换的调度方案中进行检索, 即可找到全局最优调度方案.

令:

$$x_{ij} = \begin{cases} 1, & M_i \text{ 承担 } T_j, i = 3, 4, \dots, m; j = 1, 2, \dots, n \\ 0, & \text{否则} \end{cases} \tag{2}$$

$$y_{ij} = \begin{cases} 1, & M_i \text{ 在 } t \text{ 时刻前承担 } T_j, i = 1, 2; j = 1, 2, \dots, n \\ 0, & \text{否则} \end{cases} \tag{3}$$

$$y'_{ij} = \begin{cases} 1, & M_i \text{ 在 } t \text{ 时刻后承担 } T_j, i = 1, 2; j = 1, 2, \dots, n \\ 0, & \text{否则} \end{cases} \tag{4}$$

其含义如图 2 所示.

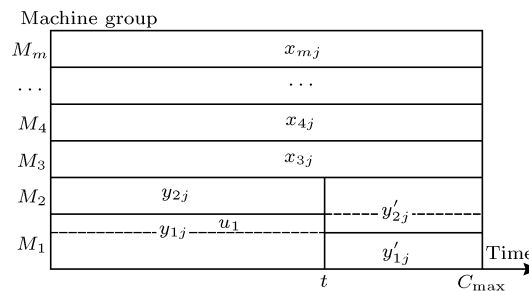


图 2  $t$  时刻  $u_1$  在  $M_1$  和  $M_2$  之间切换  
Fig. 2  $u_1$  shift to  $M_2$  from  $M_1$  at time  $t$

因此, 建立如下的 0-1 整数规划模型:

$$\begin{aligned} & \min C_{\max} \\ & \text{s.t. } \sum_{j=1}^n \max \left[ p_{ij}, \frac{1}{\sum_{u_l \in U_i} (1/q_{lj})} \right] x_{ij} - C_{\max} \leq 0, \quad i = 3, 4, \dots, m \end{aligned} \quad (5)$$

$$\sum_{j=1}^n \max \left[ p_{ij}, \frac{1}{\sum_{u_l \in U_i^-} (1/q_{lj})} \right] y_{ij} - t \leq 0, \quad i = 1, 2 \quad (6)$$

$$\sum_{j=1}^n \max \left[ p_{ij}, \frac{1}{\sum_{u_l \in U_i^+} (1/q_{lj})} \right] y'_{ij} - (C_{\max} - t) \leq 0, \quad i = 1, 2 \quad (7)$$

$$\sum_{i=3}^m x_{ij} + \sum_{i=1}^2 y_{ij} + \sum_{i=1}^2 y'_{ij} = 1, \quad j = 1, 2, \dots, n \quad (8)$$

$$x_{ij} \in \{0, 1\}, \quad i = 3, 4, \dots, m; \quad j = 1, 2, \dots, n \quad (9)$$

$$y_{ij}, y'_{ij} \in \{0, 1\}, \quad i = 1, 2; \quad j = 1, 2, \dots, n \quad (10)$$

$$t \geq 0 \quad (11)$$

$$C_{\max} \geq 0 \quad (12)$$

其中  $p_{ij}$  表示主机  $M_i$  完成任务  $T_j$  的时间,  $q_{lj}$  表示从机  $u_l$  完成任务  $T_j$  的时间, 均为预先确定的常数.  $U_i$  表示主机  $M_i$  连接的从机集合, 对于  $i = 1, 2$ ,  $U_i^-, U_i^+$  分别表示  $M_i$  在  $t$  时刻前后连接的从机集合. (5) 使除  $M_1, M_2$  外的其他机器, 在  $C_{\max}$  之内完成加工任务; (6) 使  $M_1, M_2$  在第一阶段承担的任务在  $t$  时刻前完成; (7) 使  $M_1, M_2$  在第二阶段承担的任务从  $t$  时刻开始, 在  $C_{\max}$  前完成; (8), (9), (10) 使所有任务只被分配一次.

这一模型的 0-1 整数变量有  $nm + n$  个, 实数变量 2 个, 约束条件  $n + m + 2$  个.

对于  $GFD = 3$ , 除一台 3 柔性度从机以外, 均为刚性从机, 因此可以采用与  $GFD = 2$  类似的方法进行调度, 经最多 1 或 2 次切换即可得到最优解. 在 (5)~(12) 算法中只需要增加 1 个  $t$  变量, 以及相应的  $y_{ij}, y'_{ij}$  即可.

#### 4.2 高柔性度主从微结构

对于  $GFD$  较大的情况, 当  $GFD$  是素数时, 均只包含一台多柔性度从机, 理论上来说, 可以采用低柔性的方法来处理. 然而事实上由于整数规划计算复杂度很高, 对于  $GFD$  较大的情况其运行时间将指数增长. 而当  $GFD$  为合数时, 尚没有简单有效的整数规划方法. 本文对于  $GFD$  较大的问题, 仅针对可抢先的 (preemptive) 情况, 提出以下线性规划方法.

将所有的从机属于的设备群进行组合, 其组合数恰为总柔性度  $GFD$ . 由于可抢先, 所以对任何一个调度方案, 总可以进行适当的调整, 使其相同微结构的时间片紧邻.

将时间划分为  $GFD$  个片断  $S = \{S_1, S_2, \dots, S_{GFD}\} = \{S_k\}_{k=1}^{GFD}$ , 每一片断对应一种微结构, 其时间跨度为变量  $t_k$ . 设时间片  $k$  中  $M_i$  的从机集合为  $U_{ik}$ ,  $i = 1, 2, \dots, m$ ;  $k = 1, 2, \dots, GFD$ . 调度问题的优化目标, 就是求每一时间片的时间跨度  $t_k$ , 以及在时间片  $k$  中设备群  $M_i/U_{ik}$  先后承担的任务  $T_j$  的比例  $x_{ijk}$ , 使总时间跨度  $C_{\max}$  最小.

据此建立如下的线性规划:

$$\begin{aligned} \min & C_{\max} \\ \text{s.t.} & \sum_{k=1}^{GFD} \sum_{i=1}^m x_{ijk} - 1 = 0, \quad j = 1, 2, \dots, n \end{aligned} \quad (13)$$

$$\sum_{j=1}^n \max \left[ p_{ij}, \frac{1}{\sum_{u_l \in U_{ik}} (1/q_{lj})} \right] x_{ijk} - t_k \leq 0, \quad i = 1, 2, \dots, m; k = 1, 2, \dots, GFD \quad (14)$$

$$\sum_{k=1}^{GFD} t_k - C_{\max} = 0 \quad (15)$$

$$x_{ijk} \geq 0, \quad i = 1, 2, \dots, m; j = 1, 2, \dots, n; k = 1, 2, \dots, GFD \quad (16)$$

$$t_k \geq 0, \quad k = 1, 2, \dots, GFD \quad (17)$$

$$C_{\max} \geq 0 \quad (18)$$

其中 (13) 保证每一任务都恰被执行完成, (14) 使每一时间片各项任务的完成时间, 不超过该时间片的时间跨度, (15) 使总作业时间等于各时间片的时间跨度之和.

对于该线性规划的最优结果, 按照时间片逐一安排每项任务, 即可形成完整的调度方案. 时间片的先后顺序及时间片内任务的加工顺序不影响优化结果. 该线性规划变量个数为  $(mn + 1)GFD + 1$ , 当  $m$  和  $GFD$  一定的情况下, 其规模为  $O(n)^{[11]}$ .

## 5 利用柔性微结构改进调度结果

上述柔性微结构下的线性规划算法, 使利用柔性微结构对调度问题进行改进提供了可能. 以下用文献 [10] 的数据集来验证这一方法.

考虑两个设备群构成的调度问题, 每一设备群由 5 台设备所组成. 为获得柔性微结构对于调度问题的影响, 对比刚性微结构 ( $GFD = 1$ ) 和柔性度最小的柔性微结构 ( $GFD = 2$ ). 柔性微结构下的调度结果  $C'_{\max}$  采用上述 0-1 规划方法计算, 实验数据集以及刚性微结构下的 LPT 算法数据则来自文献 [10], 调度问题实例样本数取 100. 作为对比, 采用穷举法求出刚性微结构下的最优调度结果  $C_{\max}$ . 全部结果列于表 2 中.

表 2 柔性微结构改进效果  
Table 2 Optimized result by flexible microstructure

任务数	设备群数	任务时间分布区间	实验结果				实验结果对比	
			刚性结构 ( $GFD = 1$ )		柔性结构 ( $GFD = 2$ )		平均 $C'_{\max}/C_{\max}$	最小 $C'_{\max}/C_{\max}$
$n$	$m$		LPT 算法	$C_{\max}$	平均 $C'_{\max}$	最小 $C'_{\max}$		
6	3	[1,20]	22.53118	22.53118	21.09139	18.77591	0.93754	0.83333
9	3	[1,20]	30.82401	30.82401	30.48363	28.16544	0.98894	0.91375
6	3	[20,50]	76.68288	73.03131	70.90555	64.87371	0.97128	0.88830
9	3	[20,50]	108.39184	105.2348	104.5323	99.79732	0.99332	0.94833
8	4	[1,20]	24.24534	22.24343	20.85619	18.53567	0.93972	0.83331
12	4	[1,20]	31.12127	31.12127	30.49198	23.95622	0.97962	0.76977
8	4	[20,50]	76.97813	74.01743	71.86189	66.93618	0.97143	0.90433
12	4	[20,50]	107.4480	105.3412	104.7442	102.62340	0.99434	0.97420
10	5	[1,20]	23.33781	22.44020	21.22435	18.68304	0.94853	0.83257
10	5	[20,50]	78.81735	74.35599	72.38549	68.26103	0.97381	0.91803

可以看出: 采用柔性结构后, 调度结果显著优于刚性结构的最优值, 平均性能提高比例在 1.1%~6.2% 之间, 个别案例能够提高 23%, 更显著优于采用 LPT 算法得出的最优解.

由于调度结果单调相关于柔性度, 因此在更高柔性度的情况下, 调度结果将显然更优, 实验数据从略.

调节设备群的规模, 柔性结构和刚性结构的优化结果对比有显著的变化. 图 3 显示了调度结果随设备群规模的变化情况.

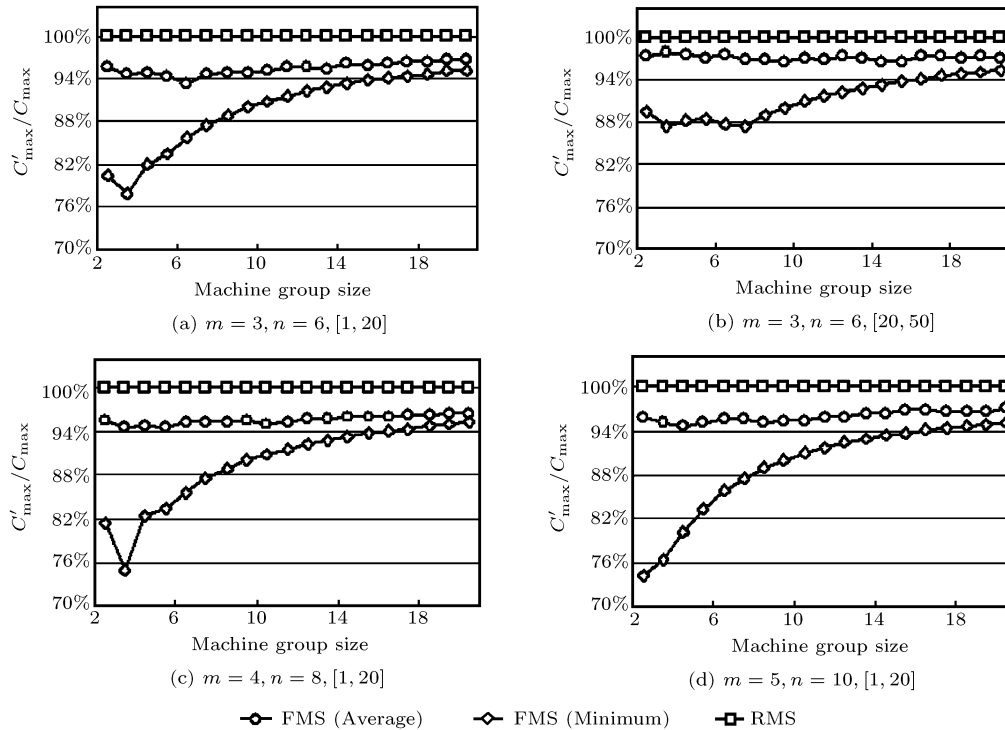


图 3 调度结果与设备群规模的关系

Fig. 3 Relation between solution and machine group size

从图 3 中可以看出, 柔性微结构能够改善调度问题最优解 4% 左右. 随着设备群规模的上升, 平均改善度基本保持不变, 但最大改善度不断削弱, 表明设备群的规模保持在适中的水平 (2~6 台/组) 是比较理想的.

## 6 结论

微结构方法在表达复杂调度问题方面具有表述严密、适应性强的特点. 本文采用微结构方法对一类新的多处理机调度问题, 即主从式多处理机调度问题进行了柔性度分类, 给出的两种线性规划方法分别适用于低柔性度和高柔性度的情况. 定量实验表明, 适当增加柔性度, 能够显著改善调度问题的平均最优解, 而适当的设备群规模 (对于 2 设备群问题, 这一合理性规模在 2~6 之间) 能够实现改善度的最大化, 这就为生产车间的设备布局提供了定量依据.

## References

- 1 Drozdowski M. Scheduling multiprocessor tasks – an overview. *European Journal of Operational Research*, 1996, **94**(2): 215~230
- 2 Jansen K, Porkolab L. Improved approximation schemes for scheduling unrelated parallel machines. *Mathematics of Operations Research*, 2001, **26**(2): 324~338

- 3 Bianco L, Błażewicz J, Dell'Olmo P, Drozdowski M. Linear algorithm for preemptive scheduling of multiprocessor tasks subject to minimal lateness. *Discrete Applied Mathematics*, 1997, **72**(1): 25~46
- 4 Angelelli E, Nagy Á B, Speranza M G, Tuza Zs. The online multiprocessor scheduling problem with known sum of the tasks. *Journal of Scheduling*, 2004, **7**(6): 421~428
- 5 Jansen K. Preemptive scheduling with dedicated processors, applications of fractional graph coloring. *Journal of Scheduling*, 2004, **7**(1): 35~48
- 6 Zhai Q Z, Guan X H, Guo Y, Sun L, Fan W. A new method for production scheduling with hybrid dynamic constraints. *Acta Automatica Sinica*, 2004, **30**(4): 539~546
- 7 Xia W J, Wu Z M. Hybrid particle swarm optimization approach for multi-objective flexible job-shop scheduling problems. *Control and Decision*, 2005, **20**(2): 137~141
- 8 Kusakabe S, Iio K, Taniguchi H, Amamiya M. Fine-grained system-call scheduling in CEFOS on commodity processors. In: *Parallel Processing and Applied Mathematics, Lecture Notes in Computer Science*, 3019. Czesochowa: Springer-Verlag GmbH, 2004. 383~388
- 9 Loughlin C. Flexible scheduling. *Assembly Automation*, 2002, **22**(3): 202
- 10 Błażewicz J, Ecker K H, Peack E, Schmidt G, Weglarz J. *Scheduling Computer and Manufacturing Processes*. 2nd ed. Berlin: Springer, 2001. 317~366, 141
- 11 Karmarkar N. A new polynomial-time algorithm for linear-programming. *Algorithmica*, 1984, **4**(4): 373~395

**沈益民** 博士研究生, 教授, 研究方向为企业建模方法、 workflow 技术、应用数学和软件工程。

(**SHEN Yi-Min** Ph. D. candidate, professor. His research interests include enterprise modeling methods, workflow technology, applied mathematics, and software engineering.)

**范玉顺** 教授, 博士生导师, 研究方向为 workflow 技术与企业经营过程重组、系统集成与集成平台、企业建模与仿真优化、网络化制造、面向对象与柔性软件系统。

(**FAN Yu-Shun** Professor, Ph. D. His research interests include workflow technology, business process reengineering, system integration and integrated platform, enterprise modeling methods and optimization analysis, networking manufacturing, object-oriented technologies, and flexible software systems.)