

文章编号: 1001-0920(2005)03-0257-04

## 一类单机动态调度问题的改进滚动时域方法

王冰<sup>1,2</sup>, 席裕庚<sup>1</sup>, 谷寒雨<sup>1</sup>

(1. 上海交通大学自动化研究所, 上海 200030; 2. 山东大学威海分校信息工程学院, 山东威海 264209)

**摘要:** 将滚动时域方法用于一类单机动态调度问题, 对其调度子问题进行改进. 在子问题的目标函数中增加一个终端惩罚项, 在优化局部目标的同时兼顾全局目标, 证明了已有分枝定界算法经适当修正可用于求解改进的调度子问题. 仿真计算结果表明, 在绝大多数情况下, 改进滚动时域方法的调度质量优于最好的规则调度方法.

**关键词:** 滚动时域方法; 调度子问题; 终端惩罚; 分枝定界算法

**中图分类号:** TH2 **文献标识码:** A

## An improved rolling horizon procedure for single-machine scheduling with release times

WANG Bing<sup>1,2</sup>, XI Yugen<sup>1</sup>, GU Hanyu<sup>1</sup>

(1. Institute of Automation, Shanghai Jiaotong University, Shanghai 200030, China; 2. School of Information Engineering, Shandong University at Weihai, Weihai 264209, China. Correspondent: WANG Bing, Email: wangbing@sdu.edu.cn)

**Abstract:** A rolling horizon procedure (RHP) is used in solving a kind of single-machine dynamic scheduling problems with release times. An improved sub-problem for the RHP is proposed. A terminal penalty function is appended to the objective of each sub-problem, in which both the local objective and the global objective are considered. It is proved that an existing branch and bound can be revised to solve the improved sub-problem. Computational results demonstrate that the improved RHP performs better than the best dispatching rule on most cases.

**Key words:** rolling horizon procedure; scheduling sub-problem; terminal penalty; branch and bound

### 1 引言

调度问题的一个突出特点是具有很强的计算复杂性。 $1/r_i/C_i$ 表示已知工件 $i$ 的到达时间 $r_i$ 和加工时间 $p_i$ , 在单台机器上安排工件加工次序和开始加工时间, 并最小化所有工件的完成时间 $C_i$ 之和. 这是强NP-hard问题<sup>[1]</sup>, 用精确算法如分枝定界法<sup>[2]</sup>, 只能对中小规模问题的离线调度求解; 随着问题规模的增大, 通常采用启发式算法. 规则调度是生产实践中广泛应用的启发式方法, 常用的规则有ECT<sup>[2]</sup>, SPT<sup>[3]</sup>, PRTF和APRTF<sup>[4]</sup>等. 规则调度计算代价小, 但其近视性将导致调度质量有时很差. 滚

动时域方法(RHPs)是一类通用的启发式方法<sup>[6-8]</sup>, 它避免了规则调度只看一步的近视特性, 在决策时刻可以预测更大时域上的工件信息.

文献[5]在全局工件信息完全的情形下, 设计了基于初始调度的滚动时域调度策略, 既利用了完全的已知信息, 又保留了滚动时域方法的计算有效性. 文献[6]在全局信息不完全的情形下, 研究了RHPs的在线调度问题, 在一般情况下虽然多付出了一些计算代价, 但调度质量优于最好的规则, 当工件到达较快时, RHPs在多数问题中调度质量较差.

本文在全局信息不完全的在线情形下, 设计出改进的滚动时域方法. 借鉴文献[5]中调度子问题

收稿日期: 2004-05-31; 修回日期: 2004-08-12

基金项目: 国家自然科学基金项目(60274013); 山东大学青年科学基金项目(11010053187075).

作者简介: 王冰(1966—), 女, 山东莱芜人, 副教授, 博士生, 从事组合优化、调度算法的研究; 席裕庚(1946—), 男, 上海人, 教授, 博士生导师, 从事预测控制、动态大系统等研究.

的设计,针对已有RHPs的弊端,改进局部调度子问题的定义,在局部目标中加入对完成时间后推的惩罚,以牺牲局部最优换取全局调度性能的改善

## 2 改进滚动时域调度方法

滚动时域调度方法将全局调度问题在时域上分解为滚动进行的多个调度子问题,每个子问题只利用当前预知的局部信息.在每一决策时刻,进入子问题进行局部调度的工件集合称为该时刻的滚动窗口.求解滚动窗口内的局部调度子问题,执行这一局部调度中的前几个工件.进入下一决策时刻,重复上述步骤,直到全部工件调度加工完毕

RHPs的滚动参数以 $(x, y, z)$ 表示,其中: $x$ 表示滚动窗口中未来工件的个数,保证局部调度对未来的预测性,在一定程度上避免了规则调度的近视性.过多的当前已到达工件只会增加调度子问题的计算负担,而不会对改善全局调度质量发挥太大作用,所以滚动窗口中已到达工件通常控制为不超过 $y$ 个. $(x, y)$ 是基于工件数目的滚动窗口的一种选择方式,控制滚动窗口的规模便控制了每次局部调度的计算量. $z$ 称为滚动调度的步长,表示每步局部调度后要执行加工的工件个数.文献[6]中RHPs滚动参数的推荐值为 $(12, 5, 2)$ ,这样选取的滚动参数,可使RHPs对调度性能和计算代价具有较好的折衷.本文采用上述滚动窗口的选择方式

$1/r_i / C_i$ 的调度子问题是以最小化工件完成时间之和为目标,所得局部调度并不一定有最小的完成时间.如果调度子问题只考虑局部最优,则其结果中插入的人为空闲时间会造成后续工件开始加工时间的推迟,从而使全局目标有增大的趋势.为克服这一缺陷,需要对调度子问题进行改进

本文对滚动时域中的调度采用如下定义和符号:在所讨论的单机调度问题的滚动时域中,全局问题的调度针对全体工件集合 $N$ ,部分调度建立在 $N$ 中的部分工件集合上;决策时刻 $t$ 表示第 $t$ 次局部调度的时刻,而不是该次决策的时间;如以 $K(t)$ 表示 $t$ 时刻的滚动窗口工件集,则以 $S(K(t))$ 表示 $K(t)$ 上的任一调度;在滚动窗口之外的未调度工件的集合称为该时刻滚动窗口的后续工件集,以 $\tilde{K}(t)$ 表示, $|\tilde{K}(t)|$ 表示 $\tilde{K}(t)$ 中的工件数目

在任一决策时刻 $t$ ,对滚动窗口内的工件先按SPT规则<sup>[3]</sup>进行排序,SPT规则在决策时刻选当前已到达工件中最小者,所以调度结果中不会插入人为空闲时间,具有可行调度中最小的完成时间 $C_{K(t)}^s$ .从局部调度的角度希望滚动窗口内性能最优,从全局角度希望局部调度具有最小的完成时间.改

进的滚动时域方法的调度子问题就是从解决二者的矛盾入手,对二者进行折衷.局部调度解 $S(K(t))$ 的完成时间 $C_{K(t)}$ 相对 $C_{K(t)}^s$ 的推迟进行惩罚.在决策时刻 $t$ ,按如下优化指标:

$$\min J_{K(t)} = \min_{S(K(t))} \left[ \sum_{i \in K(t)} C_i + |\tilde{K}(t)| \Delta C_{K(t)} \right] \quad (1)$$

对滚动窗口 $K(t)$ 中的工件进行最优调度.其中

$$\Delta C_{K(t)} = \max\{0, C_{K(t)} - C_{K(t)}^s\}, \quad (2)$$

$$|\tilde{K}(t)| = n - (t-1)z - |K(t)| \quad (3)$$

在优化指标(1)中,第1项 $\sum_{i \in K(t)} C_i$ 为滚动窗口子

问题的局部 $C_i$ 性能指标,是全局性能指标在滚动窗口部分调度中的体现;第2项 $|\tilde{K}(t)| \Delta C_{K(t)}$ 是对滚动窗口的局部调度造成后续部分调度后推的惩罚,该惩罚对最坏的情况进行,即当前局部调度完成时间相对 $C_{K(t)}^s$ 的推后时间 $\Delta C_{K(t)}$ ,会造成后续所有工件的开始加工时间后推 $\Delta C_{K(t)}$ ,反映局部调度兼顾了对全局调度的影响.该调度子问题的求解须假定在初始时刻已知所有工件的总数为 $n$ ,在实际调度中这一要求是容易实现的

本文将滚动窗口内在式(1)优化指标下的局部调度子问题称为改进的调度子调度,将具有这样的局部调度子问题的滚动时域方法(RHPs)称为改进的滚动时域方法(IRHPs).

## 3 改进调度子问题的分枝定界算法

对于子问题(1)这种带终端惩罚的优化指标,采用文献[2]中分枝定界法的优越关系,可证明下述3个定理所述的优越关系在本文改进调度子问题中仍然成立

由于在调度子问题中决策时刻固定,调度子问题的算法与滚动调度决策时刻 $t$ 无关,为表述方便,本文将 $t$ 隐去,采用与滚动时域中不同的记号.滚动窗口工件集合以 $K$ 表示, $K$ 上的完整调度以 $S_K$ 表示;滚动窗口中任一调度的完成时间以 $C_K$ 表示,SPT规则调度的完成时间以 $C_K^s$ 表示;如果 $K$ 上的特定调度以 $S_K^i$ 表示,则其开始时间和完成时间以 $B_K^i$ 和 $C_K^i$ 表示; $S_K$ 在式(1)下的优化指标以 $J(S_K)$ 表示,其第1项 $\sum C_i$ 性能以 $J^R(S_K)$ 表示,第2项惩罚以 $J^T(S_K)$ 表示; $K$ 中的部分工件集 $Y$ 的调度以 $S_Y$ 表示, $Y$ 上的某一特定调度以上标注为 $S_Y$ 中某一工件 $i$ 的开始加工时间和加工完成时间以 $B_i(S_Y)$ 和 $C_i(S_Y)$ 表示, $S_Y$ 调度中第 $y$ 个位置上工件 $i$ 的开始加工时间和加工完成时间以 $B_{\bar{y}}(S_Y)$ 和 $C_{\bar{y}}(S_Y)$ 表示;部分调度 $S_Y$ 接工件 $i$ 的调度以 $(S_Y, i)$ 表示; $Y$ 关于 $K$ 的补集以 $\bar{Y}$ 表示, $\bar{Y}$ 的相应表示与 $Y$ 类似;部分调度 $S_Y$ 和 $S_{\bar{Y}}$ 连接组成调度 $S_K = (S_Y, S_{\bar{Y}})$ ;对 $K$ 上

的某一特定调度  $S_k^i = (S_Y, S_{\bar{Y}}^i)$ , 有  $J^R(S_k^i) = C_j(S_k^i), J^T(S_k^i) = |\tilde{K}| \Delta C_k^i$ , 则式(1)的优化指标为  $J(S_k^i) = J^R(S_k^i) + J^T(S_k^i)$ . 其中:  $J^R(S_k^i) = J^R(S_Y) + J^R(S_{\bar{Y}}^i), \Delta C_k^i = \Delta C_{\bar{Y}}^i = C_{\bar{Y}}^i - C_k^S$ .

**定理 1** 对于工件集合  $K$  和其中  $y$  个工件的部分调度  $S_Y (Y \subset K)$ , 如果工件  $i \in \bar{Y} = K - Y$  与所有工件  $j \in \bar{Y}$  有  $p_i < p_j$ , 且对任一工件  $h \in \bar{Y}$ , 有  $B_h(S_Y) < B_i(S_Y)$ , 则在位置  $y + 1$  上工件  $i$  优于工件  $h$ ; 如果  $B_h(S_Y) > B_i(S_Y)$ , 则  $(S_Y, i)$  严格优超  $(S_Y, h)$ .

**证明** 由文献[2]中定理 1 的证明知, 排序  $S_k^h = (S_Y, S_{\bar{Y}}^h)$  表示  $K$  中第  $y + 1$  个位置是工件  $h$  的优化调度, 而工件  $i$  是在第  $x$  个位置上的排序,  $x > y + 1$ . 由于  $r_i < B_i(S_Y) < B_h(S_Y) = B_{y+1}(S_Y) < B_m(S^h), y < m < x$ , 交换工件  $i$  与它前面位置  $x - 1$  上的工件, 不会增加位置  $x - 1$  和  $x$  的工件完成时间, 也不会增加其后工件的完成时间. 继续将  $i$  与它前一个工件位置进行类似交换, 直到  $i$  交换到位置  $y + 1$  上, 这样产生的排序为  $S_k^i = (S_Y, S_{\bar{Y}}^i)$ .  $S_k^i$  的第  $y + 1$  个位置上是工件  $i$ , 而工件  $h$  在  $y + 1$  之后,  $S_k^i$  中任一位置上工件的完成时间都不大于  $S_k^h$  中相应位置上工件的完成时间, 所以  $J^R(S_k^i) < J^R(S_k^h)$ , 且  $C_k^i < C_k^h, \Delta C_k^i < \Delta C_k^h$ . 则  $J^T(S_k^i) < J^T(S_k^h)$ , 所以  $J(S_k^i) < J(S_k^h)$ .

**定理 2** 对于  $n$  个工件的集合  $K$  和其中  $y$  个工件的部分调度  $S_Y (Y \subset K)$ , 某一工件  $i \in \bar{Y}$ , 对于所有工件  $j \in \bar{Y}$ , 有  $C_i(S_Y) < C_j(S_Y)$ ; 如果  $r_j < C_i(S_Y)$ , 则部分调度  $(S_Y, j)$  被严格优超

**证明略**

**定理 3** 对于工件集合  $K$  和部分调度  $S_Y (Y \subset K)$ , 两个工件  $i, j \in \bar{Y}$ , 如果  $p_j < p_i$ , 且  $C_j(S_Y) < C_i(S_Y)$ , 则  $(S_Y, i)$  优超  $(S_Y, j)$ .

**证明** 由文献[2]中定理 3 的证明知, 排序  $S_k^j = (S_Y, S_{\bar{Y}}^j)$  表示  $K$  中第  $y + 1$  个位置是工件  $j$  的优化调度, 工件  $i$  在  $j$  之后, 将  $j$  与  $i$  的位置交换, 而其他工件的位置不变, 得到的新调度  $S_k^i$  仍是可行的. 在新调度  $S_k^i$  中, 两工件的完成时间都不大于在  $S_k^j$  中的完成时间, 则新调度  $S_k^i$  的完成时间也不大于  $S_k^j$  的完成时间. 所以  $J^R(S_k^i) < J^R(S_k^j), J^T(S_k^i) < J^T(S_k^j), J(S_k^i) < J(S_k^j)$ .

对于式(1)的惩罚性能指标, 本文调度子问题最优调度的上下界求法由下述定理给出:

**定理 4** 对于工件集  $K$ , 如果部分调度  $S_Y$  已经确定, 则可证明对于未调度工件  $\bar{Y}$ , 用 ECT 规则产生调度  $S_{\bar{Y}}^E$ , 则调度  $S_k^E = (S_Y, S_{\bar{Y}}^E)$  对于式(1)描述的

目标函数是最优目标函数的一个上界; 用 SRPT<sup>[9]</sup> 规则产生调度  $S_{\bar{Y}}^{SR}$ , 则调度  $S_k^{SR} = (S_Y, S_{\bar{Y}}^{SR})$  对于式(1)目标函数是最优目标函数的一个下界

**证明** 对于  $1/r_i / C_i$  问题, 在部分调度  $S_Y$  确定的情况下, 调度  $S_k^E = (S_Y, S_{\bar{Y}}^E)$  是工件集合  $K$  的最优调度  $S_k^* | S_Y = (S_Y, S_{\bar{Y}}^*)$  的一个上界, 即

$$J^R(S_Y, S_{\bar{Y}}^E) \geq J^R(S_Y, S_{\bar{Y}}^*). \quad (4)$$

设调度  $(S_Y, S_{\bar{Y}}^E)$  的完成时间为  $C_{\bar{Y}}^E, (S_Y, S_{\bar{Y}}^E)$  的完成时间为  $C_{\bar{Y}}^E$ . 对于工件集合  $K$ , 由于其 SPT 规则调度的完成时间  $C_k^S$  是所有可行调度中最小的, 所以  $\Delta C_k^E = C_{\bar{Y}}^E - C_k^S \geq 0, \Delta C_k^E = C_{\bar{Y}}^E - C_k^S \geq 0$  在部分调度  $S_Y$  确定的情况下, 对于部分工件集  $\bar{Y}$ , 其 ECT 规则调度的完成时间一定不小于最优调度的完成时间, 即  $C_{\bar{Y}}^E \geq C_{\bar{Y}}^*$ , 所以  $\Delta C_k^E \geq \Delta C_k^*$ ; 而  $J^T(S_Y, S_{\bar{Y}}^E) = |\tilde{K}| \Delta C_k^E, J^T(S_Y, S_{\bar{Y}}^*) = |\tilde{K}| \Delta C_k^*$ , 所以  $J^T(S_Y, S_{\bar{Y}}^E) \geq J^T(S_Y, S_{\bar{Y}}^*)$ . 结合式(4)得

$$J(S_Y, S_{\bar{Y}}^E) \geq J(S_Y, S_{\bar{Y}}^*) = J(S_k^* | S_Y).$$

文献[9]已证明 SRPT 规则调度是  $1/r_i / C_i$  问题最优调度的下界, 即

$$J^R(S_Y, S_{\bar{Y}}^{SR}) \leq J^R(S_Y, S_{\bar{Y}}^*). \quad (5)$$

设调度  $(S_Y, S_{\bar{Y}}^{SR})$  的完成时间为  $C_{\bar{Y}}^{SR}, \Delta C_k^{SR} = C_{\bar{Y}}^{SR} - C_k^S$ . 对于部分工件集  $\bar{Y}$ , 其 SRPT 规则调度的完成时间一定不大于 SPT 规则调度的完成时间, 而 SPT 规则调度的完成时间一定不大于最优调度的完成时间, 则得  $C_{\bar{Y}}^{SR} \leq C_{\bar{Y}}^*$ , 所以  $\Delta C_k^{SR} \leq \Delta C_k^*$ ; 而  $J^T(S_Y, S_{\bar{Y}}^{SR}) = |\tilde{K}| \Delta C_k^{SR}$ , 则  $J^T(S_Y, S_{\bar{Y}}^{SR}) \leq J^T(S_Y, S_{\bar{Y}}^*)$ . 综合式(5)得

$$J(S_Y, S_{\bar{Y}}^{SR}) \leq J(S_Y, S_{\bar{Y}}^*) = J(S_k^* | S_Y).$$

#### 4 仿真计算与分析

本节通过仿真实验来研究改进的滚动时域调度的全局性能, 并与规则调度和 RHP 两类在线调度方法进行比较

IRHP 的滚动参数  $(x, y, z)$  取值  $(12, 5, 2)$ , 调度子问题的算法采用第 3 节的分枝定界法, 所有方法均用 C 语言编程, 在 Windows XP 操作系统环境下, 用 Visual C++ 6.0 编译软件, 实验在 Pentium 4-M CPU 1.80 GHz 微机上. 问题用与文献[6]相似的方式随机产生. 对于工件数目为  $n$  的问题, 工件的加工时间  $p_i$  在  $1 \sim 100$  间一致分布, 工件的到达时间  $r_i$  由  $0 \sim UL$  间的一致分布产生,  $UL = 50.5n\rho$ . 其中  $\rho$  是表征工件到达快慢的参数,  $\rho$  的取值通常为 10 个, 本文取  $0.2 \sim 3.0$ .

对于每一随机产生的问题, 取 3 个规则 SPT, PRTF 和 APTF 中调度性能最好者, 表示为 SPA. 同一问题先用 3 种规则调度, 得到 SPA 的解, 再用

**IRHP 求解** 如果 RHP 的性能好于 SPA, 则相对比较值用  $(SPA - IRHP)/RHP$  计算; 如果 SPA 的性能好于 IRHP, 则相对比较值用  $(RHP - SPA)/SPA$  计算. 测试实验针对工件数目为 50 和 250 两组进行, 共测试 400 个问题, 每组测试 200 个问题.

实验结果如表 1 和表 2 所示. 其中“数目”是 20 个问题中出现该情形的数目, “平均”是 20 个问题中该情形下的平均改善量, “最大”是 20 个问题中该情形下的最大改善量. 可以看出, 在绝大多数情况下, IRHP 的调度性能好于规则调度. 当工件规模为较小的 50 个时, IRHP 好于 SPA 的相对比较值较大, 尤其在  $\rho$  值不大于 1 时, SPA 只有个别问题好于 IRHP. 当工件规模为 250 个时, RHP 好于 SPA 的问题更普遍, 但相对比较值却变小, 说明当问题规模增大时, IRHP 与 SPA 越来越接近. 当问题规模趋向无穷时, SPT 规则是渐近最优的, 所以当问题规模越来越大时, IRHP 的绝对性能也越来越好.

表 1 IRHP 与 SPA 的对比 (50 工件问题)

$\rho$	IRHP 好于 SPA			SPA 好于 RHP		
	数目	平均/%	最大/%	数目	平均/%	最大/%
0.20	7	0.368	1.063	1	0.257	0.257
0.40	14	0.099	0.483	1	0.287	0.287
0.60	14	0.156	0.719	1	0.032	0.032
0.80	16	0.171	0.593	1	0.059	0.059
1.00	19	0.110	0.371	0	0	0
1.25	16	0.045	0.127	0	0	0
1.50	14	0.034	0.104	0	0	0
1.75	14	0.024	0.069	0	0	0
2.00	10	0.031	0.075	0	0	0
3.00	8	0.012	0.023	0	0	0

表 2 IRHP 与 SPA 的对比 (250 工件问题)

$\rho$	IRHP 好于 SPA			SPA 好于 RHP		
	数目	平均/%	最大/%	数目	平均/%	最大/%
0.20	15	0.005	0.011	5	0.007	0.013
0.40	15	0.010	0.033	4	0.009	0.028
0.60	16	0.014	0.035	4	0.025	0.055
0.80	17	0.019	0.076	3	0.030	0.037
1.00	19	0.012	0.024	1	0.015	0.015
1.25	20	0.018	0.033	0	0	0
1.50	20	0.016	0.026	0	0	0
1.75	20	0.015	0.031	0	0	0
2.00	20	0.010	0.023	0	0	0
3.00	20	0.002	0.004	0	0	0

IRHP 与 RHP 一样, 在  $\rho$  值不大于 1 的问题中存在差于 SPA 的情形; 当  $\rho > 1$  时, IRHP 和 RHP 都好于 SPA. 不同之处在于, IRHP 相对于 RHP 好于 SPA 的问题个数明显增多, 说明 IRHP 在更多的问题中表现出良好的性能.

针对 50 工件和 250 工件两组问题, 对 IRHP 与

RHP 进行比较实验. 共测试 400 个问题, 比较结果如表 3 和表 4 所示. 可以看出, 在两种规模的问题中, IRHP 一致好于 RHP, 且在  $\rho$  值小于 1.25 时, IRHP 相对 RHP 有明显的改善; 规模较大时, 这种改善稳定且明显, 而这正是 RHP 表现不佳的情形. 在  $\rho$  值较小的问题中, 局部调度与全局问题之间的耦合关系更强, 而 RHP 的子问题设计割裂了这种耦合关系, 所以 RHP 的调度性能较差. IRHP 的子问题设计考虑了局部目标与全局目标的一致性, 在 RHP 表现较差的问题中具有较大的改善.

表 3 IRHP 与 RHP 的对比 (50 工件问题)

$\rho$	IRHP 好于 RHP			RHP 好于 IRHP		
	数目	平均/%	最大/%	数目	平均/%	最大/%
0.20	20	2.05	3.72	0	0	0
0.40	20	1.20	1.34	0	0	0
0.60	20	0.28	0.93	0	0	0
0.80	20	0.14	0.21	0	0	0
1.00	20	0.09	0.11	0	0	0
1.25	20	0.10	0.27	0	0	0
1.50	20	0.03	0.28	0	0	0
1.75	0	0	0	0	0	0
2.00	0	0	0	0	0	0
3.00	0	0	0	0	0	0

表 4 IRHP 与 RHP 的对比 (250 工件问题)

$\rho$	IRHP 好于 RHP			RHP 好于 IRHP		
	数目	平均/%	最大/%	数目	平均/%	最大/%
0.20	20	0.722	1.153	0	0	0
0.40	20	0.534	1.291	0	0	0
0.60	20	0.413	0.712	0	0	0
0.80	20	0.354	0.653	0	0	0
1.00	20	0.209	0.483	0	0	0
1.25	18	0.015	0.046	2	0.008	0.012
1.50	18	0.007	0.021	1	0.005	0.005
1.75	20	0.005	0.015	0	0	0
2.00	17	0.004	0.013	1	0.002	0.002
3.00	11	0.001	0.002	0	0	0

### 5 结论

对于在调度时刻全局信息不完全的动态调度问题, 滚动时域调度方法是有效的, 其边预测、边调度、边执行的思想适合动态调度问题. 对于  $1/r_i / C_i$  的动态调度问题, 在决策时刻片面追求局部目标最优, 可能导致全局目标有增大的趋势. 在局部调度子问题的目标函数中引入终端惩罚, 可以改进由此导致的弊端. 仿真计算结果表明, 在改进的滚动时域方法中, 全局调度质量优于最好的规则调度的问题数目明显增多, 相对已有 RHPs 的调度质量有全面的改善, 特别是对 RHPs 表现较差的情形, 这种改善更加明显.

(下转第 265 页)

## 5 结 论

本文将T-S模糊模型转化为RBF神经网络,提出一种新的模糊神经网络混沌混合优化方法。首先采用混沌变换获得所需概率分布的序列,应用混沌变换序列优化获得初步优化群;然后应用FCM获得最优中心,解决了神经网络模型复杂性与逼近精度和泛化性能的矛盾,增强了模型的预测能力。实际应用结果表明,采用该方法能达到较好的建模效果,但算法较为复杂,计算量较大。如何简化计算,使算法既能较快收敛又能保持全局最优,将是今后研究的重点。

### 参考文献(References)

- [1] Bezdek J C. Convergence theory for fuzzy C-means: Counterexamples and repairs [J]. *IEEE Trans on Systems, Man and Cybernetics*, 1987, 17(2): 873-877.
- [2] Tsao E C, Bezdek J C. Fuzzy Kohonen clustering networks [J]. *Pattern Recognition*, 1994, 27(5): 757-764.
- [3] Wang L X, Medel J M. Backpropagation fuzzy system as nonlinear dynamic system identifier [A]. *IST IEEE Int Conf on Fuzzy Systems* [C]. San Diego, 1992: 1409-1418.
- [4] 张彤, 王宏伟, 王子才. 变尺度混沌优化方法及其应用 [J]. *控制与决策*, 1999, 14(3): 285-287.  
(Zhang T, Wang H W, Wang Z C. Mutative scale chaos optimization algorithm and its application [J]. *Control and Decision*, 1999, 14(3): 285-287.)
- [5] Hunt K J. Extending the functional equivalence of radial basis function networks and fuzzy inference system [J]. *IEEE Trans on Neural Networks*, 1996, (3): 776-781.
- [6] Chen M Y, Linkens D A. A fuzzy modelling approach using hierarchical neural networks [J]. *Neural Computing and Applications*, 2000, (9): 44-49.
- [7] 原隆康. 转炉终点动态磷控制技术开发 [J]. *世界钢铁*, 2000, 2: 53-59.  
(Yuan L K. Development of dynamic control technology for the endpoint phosphorus content of converter [J]. *World Iron and Steel*, 2000, 2: 53-59.)
- [8] 谢书明, 高宪文, 柴天佑. 基于灰色模型的转炉炼钢终点预报研究 [J]. *钢铁研究学报*, 1999, 11(4): 9-11.  
(Xie S M, Gao X W, Chai T Y. BOF endpoint prediction based on grey model [J]. *J of Iron and Steel Research*, 1999, 11(4): 9-11.)
- [9] 李红军, 秦永胜, 徐用懋. 化工过程中的数据协调及显著误差检测 [J]. *化工自动化及仪表*, 1997, 24(2): 25-32.  
(Li H J, Qin Y S, Xu Y M. Data reconciliation and gross error detection in chemical process [J]. *Control and Instruments in Chemical Industry*, 1997, 24(2): 25-32.)
- [10] Lin Y, Cunningham G A. A new approach to fuzzy-neural system modeling [J]. *IEEE Trans on Fuzzy Systems*, 1995, (3): 190-197.

(上接第260页)

### 参考文献(References)

- [1] Lawler E L, Lenstra J K, Rinnooy K, et al. *Sequencing and scheduling: Algorithm and complexity* [M]. Amsterdam, 1993.
- [2] Dessouky M I, Deogun J S. Sequencing jobs with unequal ready times to minimize mean flow time [J]. *SIAM J on Computing*, 1981, 10(1): 192-202.
- [3] Chandra R. On  $n/1/\sqrt{F}$  dynamic deterministic problems [J]. *Naval Research Logistics Quarterly*, 1979, 26: 537-544.
- [4] Chu C B. Efficient heuristics to minimize total flow time with release dates [J]. *Operations Research Letters*, 1992, 12: 321-330.
- [5] Wang B, Xi Y G, Gu H Y. Rolling horizon heuristic for single machine scheduling problem [A]. *Proc of 2004 World Conf of Intelligent Control and Automation* [C]. Hangzhou, 2004: 2872-2876.
- [6] Suresh Chand, Rodney Traub, Reha Uzsoy. Rolling horizon procedures for the single machine deterministic total completion time scheduling problem with release dates [J]. *Annals of Operations Research*, 1997, 70: 115-125.
- [7] Fang J, Xi Y G. Rolling horizon job shop rescheduling strategy in the dynamic environment [J]. *Int J of Advanced Manufacturing Technology*, 1997, 13: 227-232.
- [8] Ahmadi R H, Bagchi U. Lower bounds for single-machine scheduling problems [J]. *Naval Research Logistics*, 1990, 37: 967-976.