

面向多机动态调度问题的两层 Q 学习算法

王国磊, 钟诗胜, 林 琳

(哈尔滨工业大学机电工程学院, 黑龙江 哈尔滨 150001)

摘 要:对于单机动态调度问题十分有效的 Q 学习, 在多机动态调度环境下却由于缺乏全局眼光而效果欠佳, 因此提出了一种双层 Q 学习算法. 底层 Q 学习着眼于局部, 以最小化设备空闲和作业平均流经时间为目标, 学习单机调度策略; 而顶层 Q 学习则着眼于全局, 以平衡机器负载、最小化整体拖期值为目标, 学习如何分配作业到合适机器. 文中分别给出了两层 Q 学习的动作集、状态空间划分方式和奖惩函数设计, 并通过对多机动态调度问题的仿真实验表明, 提出的双层 Q 学习能够很好地解决改善动态环境下多机调度问题.

关键词:动态多机调度; Q 学习; 动作集; 状态空间划分; 奖惩函数

中图分类号:TP273 **文献标识码:**A **文章编号:**1673-4785(2009)03-0239-06

Bi-level Q-learning algorithm for dynamic multi-machine scheduling problems

WANG Guo-lei, ZHONG Shi-sheng, LIN Lin

(School of Mechanical Engineering, Harbin Institute of Technology, Harbin 150001, China)

Abstract: Traditional Q-learning is very effective in dynamic single-machine scheduling problems, yet sometimes it cannot get optimal results for dynamic multi-machine scheduling problems due to its lack of global vision. To resolve this, a two-layer Q-learning algorithm was put forward. The bottom-level of Q-learning was focused on localized targets in order to learn the optimal scheduling policy which can minimize machine idleness and the mean flow time of single machines. On the other hand, the top-level of Q-learning was focused on global targets in order to find the dispatching policy which can balance machine loads and minimize the overall tardiness of all jobs. The scheduling and dispatching rules of agents, the method for dividing state space and the reward functions were all examined. Simulation results showed that the proposed two-layer Q-learning algorithm can improve the results of dynamic multi-machine scheduling problems.

Keywords: dynamic multi-machine scheduling; Q-learning; action set; state space division; reward function

Q 学习是一种典型的强化学习方法, 它不需要建立任何领域模型, 而是直接优化一个可迭代计算的 Q 函数, 获得最优控制策略. 近年来, Q 学习得到了研究人员的广泛重视^[1], 但其应用目前还主要集中在游戏比赛、控制系统和机器人领域, 在作业排序问题上的应用尚不多见^[2]. 比较经典的研究有: Aydin 等人利用 Q-III 算法训练智能体动态选择调度规则^[3], Wang 等人将强化学习应用于动态单机调度研究^[4-5]. 国内学者中魏英姿最早将强化学习

应用于车间动态作业调度^[6], 其后王世进等人也分别对利用 Q 学习算法实时选择调度规则进行了研究^[7-8]. 这些研究表明, Q 学习能够使智能体从给定的调度规则中选择出较好的调度规则.

但是, 这些研究还都仅限于单机调度问题. 对于多于多机动态调度问题, 如果仍然将每台机器视为 Agent, 利用 Q 学习根据各自的局部目标选择调度策略, 那么往往会因为缺乏全局眼光而不能达到最优效果. 因此, 本文进一步设计了符合多机动态调度问题特点的双层 Q 学习机制, 以保证整个系统的行为能朝着预期的方向演化.

1 自适应多 Agent 动态调度模型

1.1 多机动态调度问题

动态多机调度问题可以描述为:某车间或工厂有加工设备若干台,生产作业随机到达,当某时刻调度环境变化导致需要进行动态调度时,则生成新的预调度方案,并根据新的方案生产直到新的随机事件发生或者所有加工任务结束。

为了便于描述,某车间的加工设备集合为 $\{M_1, M_2, \dots, M_m\}$, 生产作业集合为 $\{J_1, J_2, \dots, J_n\}$ 。作业 J_i 到达的时间为 T_i^A , 截止时间为 T_i^D , 拖期完成时单位时间的拖期惩罚为 C_i^T 。作业 J_i 包含 n_i 道工序, 由于实际生产中加工设备具有一定的可替换性; 因此工序 O_{ij} 有若干台可用机器集合, 在机器 M_k 上的加工时间用 l_{ijk} 表示。此外, 工序 O_{ij} 的时间要素可以用六元组 $[T_{ij}^{ss}, T_{ij}^{df}, T_{ij}^{ss}, T_{ij}^{df}, T_{ij}^{ss}, T_{ij}^{df}]$ 表示, 其中: T_{ij}^{ss} 表示由作业到达时间推算出的工序最早可能开始时间, T_{ij}^{df} 表示由作业截止时间推算出的工序最迟必须结束时间, T_{ij}^{ss} 为每次预调度中工序的开始时间, T_{ij}^{df} 为每次预调度中工序的结束时间, T_{ij}^{ss} 为工序的实际开始时间, T_{ij}^{df} 为工序的实际结束时间。

1.2 建模

针对前述多机动态调度问题, 建立了一个多 Agent 动态调度模型, 模型中包括的几种主要 Agent 及其功能如表 1 所示。

表 1 Agent 定义

Table 1 Agent definition

Agent 种类	功能
机器 Agent	以最小化设备空闲、平均流经时间为目标, 调度缓冲区内任务
分配 Agent	以平衡设备使用、最小化作业的整体拖期值为目标, 分配作业到合适机器
作业 Agent	管理作业, 根据报完工情况和设备故障情况调度后续工序进入全局缓冲

如图 1 所示, 这个动态调度模型的工作原理是: 如果某时刻有新作业到来, 则动态生成非永久型的作业 Agent (job Agent, JA), JA 将作业的首工序添加到全局缓冲中; 分配 Agent (dispatching Agent, DA) 感知到全局缓冲中有工序进入则立即开始调度, 根据某种规则将作业分配到合适机器的缓冲中;

机器 Agent (machine Agent, MA) 按照某种优先规则对缓冲中的作业进行加工, 并且每完成一道工序都要向 JA 报完工, 以便 JA 将其后续工序添加到全局缓冲中; 如果发生设备故障等异常, MA 将其缓冲中待加工作业退回给 JA, 由 JA 将其置于全局缓冲中重新分配。

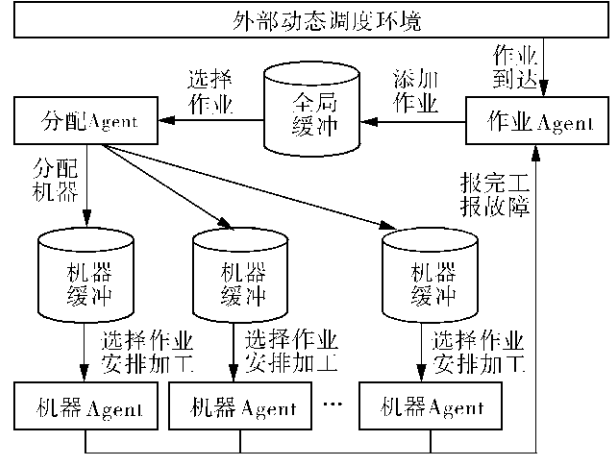


图 1 基于 Q 学习的多 Agent 动态调度机制

Fig. 1 Q-learning based multi-agent dynamic scheduling mechanism

2 双层强化学习机制

针对上述调度机制, 提出了一种基于 Q 学习的双层强化学习机制, 用于指导 DA 的机器分配和 MA 的作业选择。

2.1 Q 学习算法

Q 学习算法最早由 Watkins 在 1989 年提出, 是目前最有效的模型无关强化学习算法之一^[9], 其基本形式如式 (1) ~ (2) 所示:

$$Q(s_i, a_i) = r(s_i, a_i) + \gamma \sum_{s_{i+1} \in S} P_{s_i, s_{i+1}}(a_i) \cdot \max_b (Q(s_{i+1}, b)), \quad (1)$$

$$Q(s_i, a_i) = (1 - \alpha) Q(s_i, a_i) + \alpha (r(s_i, a_i) + \gamma \max_b (Q(s_{i+1}, b))). \quad (2)$$

式中: $Q(s_i, a_i)$ 表示 Agent 在状态 s_i 下采用动作 a_i 所获得的总计期望奖惩, 也称为状态-动作对值; $r(s_i, a_i)$ 表示 Agent 在状态 s_i 下采用动作 a_i 所获得的即时奖惩; $P_{s_i, s_{i+1}}(a_i)$ 表示在状态 s_i 下采用动作 a_i 转入状态 s_{i+1} 的概率; γ 为未来奖惩的折扣系数, 有 $0 \leq \gamma \leq 1$, γ 越接近于 0, Agent 越不考虑长远, 更趋于接收即时奖惩, γ 越接近于 1, Agent 则越具有远见, 能减少即时奖惩对学习策略的影响; b 为状态

s_{i+1} 下可采取的动作;显然 $\max_b(Q(s_{i+1}, b))$ 表示的是状态 s_{i+1} 下采用不同动作所能得到的最优奖励值. Q 学习的步骤如表 2 所示.

表2 Q 学习步骤

Table 2 Steps of Q-learning

步 骤	方 法
1	初始化聚类中心向量以及所有状态-动作对的 Q 值.
2	观察当前系统状态 s_t .
3	根据 ϵ -greedy 策略选择动作 a_t .
4	执行动作 a_t , 观察奖惩值 $r(s_t, a_t)$ 和一个状态 s_{t+1} .
5	根据式(2)更新状态-动作对值.
6	令 $s_t = s_{t+1}$ 并返回步骤 2, 直到该 episode 结束.
7	重复执行一定数量的 episode, 直到学习过程完成.

2.2 底层强化学习

底层 Q 学习的目的是使 MA 能够根据缓冲区状态选择最优的调度策略,从而达到最小化机器空闲和作业平均流经时间的局部目标.

2.2.1 动作集

为此,文中选用了常见的 3 种调度规则作为设备 Agent 的动作集,如表 3 所示.

表3 设备 Agent 的动作集

Table 3 Action set of equipment agent

规 则	说 明
EDD (earliest due date)	最小交货期原则,即选择缓冲区内具有 $\min(T_i^d)$ 的作业
SPT (shortest processing time)	最短加工时间原则,即选择缓冲区内具有 $\min(l_i)$ 的作业
FIFO (first in first out)	先到先服务原则,即选择缓冲区内具有 $\min(T_i^m)$ 的作业

2.2.2 状态划分

状态空间的划分是合理选择调度规则的基础,状态范围的划分应尽可能细化,且尽可能使系统所处的状态在各个范围分布比较均衡. 通过大量仿真试验,本文确定了基于两种特性指标的状态划分策略,如图 2 所示,共分为 18 种状态. 此外,底层 Q 学

习还有 2 种虚状态,即缓冲区内没有作业或者只有一道作业,这种情况下无须进行调度规则的选取.

1) 设备利用率因子 F_i^{UR} . 即 t 时刻前设备占用时间和总可用时间的比值,如式(3)所示.

$$F_i^{UR} = t_u / t. \quad (3)$$

式中: t_u 表示 t 时刻前设备的有效利用时间.

2) 作业紧张程度因子 F_i^{EMT} . 即 t 时刻缓冲区内作业的平均拖期期望,如式(4)所示.

$$F_i^{EMT} = \frac{1}{N_b} \sum_{i=1}^{N_b} ((t + l_i) - T_i^d). \quad (4)$$

式中: N_b 表示缓冲区内作业数目. 而每个状态的范围则由缓冲区内作业平均加工时间

$$T_i^{MP} = \frac{1}{N_b} \sum_{i=1}^{N_b} l_i \quad (5)$$

乘以系数 x 确定.

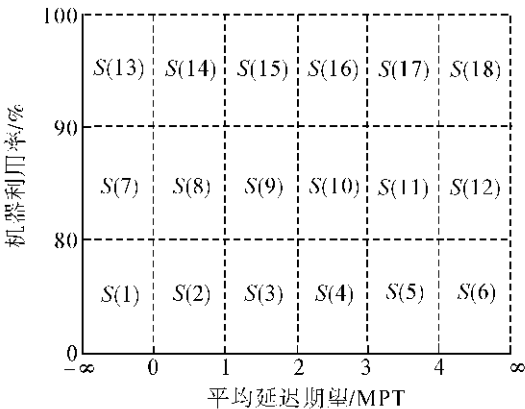


图2 底层 Q 学习的状态空间划分

Fig.2 State space division of bottom level Q-learning

2.2.3 奖惩函数设计

综合考虑设备利用率和作业拖期 2 项因素,奖励函数设计如表 4 所示. 以奖励类别 1 为例,其含义为如果当前状态 s_t 的设备利用率小于 0.8,并且 Agent 动作使缓冲区内剩余工序的平均延迟期望增大,设备利用率提高,那么该动作的立即奖惩为 5.

表4 底层 Q 学习奖惩函数

Table 4 Reward function of bottom level Q-learning

奖励类别	判别标准		奖 惩
1	$F_i^{UR} < 0.8$	$F_i^{UR} \nearrow, F_i^{EMT} \nearrow$	$r = 5$
2	$F_i^{UR} < 0.8$	$F_i^{UR} \nearrow, F_i^{EMT} \searrow$	$r = 10$
3	$F_i^{UR} < 0.8$	$F_i^{UR} \searrow, F_i^{EMT} \nearrow$	$r = -10$
4	$F_i^{UR} < 0.8$	$F_i^{UR} \searrow, F_i^{EMT} \searrow$	$r = -5$
5	$0.8 \leq F_i^{UR} < 0.9$	$F_i^{UR} \nearrow, F_i^{EMT} \nearrow$	$r = -1$
6	$0.8 \leq F_i^{UR} < 0.9$	$F_i^{UR} \nearrow, F_i^{EMT} \searrow$	$r = 10$

续表 4

奖励类别	判别标准	奖 惩
7	$0.8 \leq F_i^{UR} < 0.9$ $F_i^{UR} \searrow, F_i^{EMT} \nearrow$	$r = -10$
8	$0.8 \leq F_i^{UR} < 0.9$ $F_i^{UR} \searrow, F_i^{EMT} \searrow$	$r = 1$
9	$0.9 \leq F_i^{UR}$ $F_i^{UR} \nearrow, F_i^{EMT} \nearrow$	$r = -5$
10	$0.9 \leq F_i^{UR}$ $F_i^{UR} \nearrow, F_i^{EMT} \searrow$	$r = 10$
11	$0.9 \leq F_i^{UR}$ $F_i^{UR} \searrow, F_i^{EMT} \nearrow$	$r = -10$
12	$0.9 \leq F_i^{UR}$ $F_i^{UR} \searrow, F_i^{EMT} \searrow$	$r = 5$

2.3 顶层强化学习

由于设备可替换性的存在, DA 的主要任务是将全局缓冲中的任务分配到合适的机器, 因此顶层 Q 学习的设计就是为了使 DA 能够根据当前作业的特征以及各可用机器的负荷合理选择加工机器。

2.3.1 动作集

DA 的动作集采用表 5 所示的 3 种调度规则。

表 5 分配 Agent 的动作集

Table 5 Action set of dispatching agent

规 则	说 明
ML (minimum load)	最小负载原则, 即选择缓冲区内作业最少的机器
SPT (shortest processing time)	最短加工时间原则, 即选择使任务加工时间最短的机器
MFR (minimum failure rate)	最小故障率原则, 即选择故障率最低的机器

2.3.2 状态划分

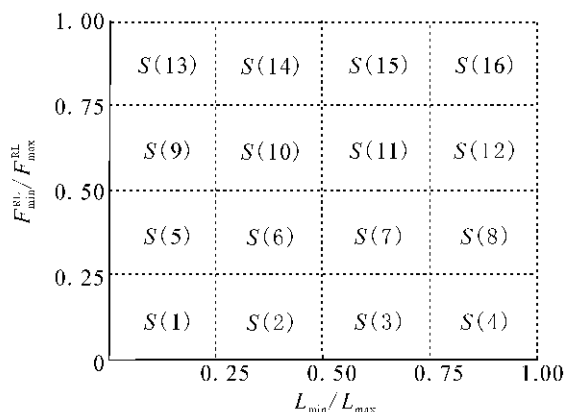


图 3 顶层 Q 学习的状态空间划分

Fig. 3 State space division of top level Q-learning

如图 3 所示, 顶层 Q 学习的状态空间以 L_{\min}/L_{\max} 和 $F_{\min}^{RL}/F_{\max}^{RL}$ 为横纵坐标划分为 16 种, 其中: $F_{\min}^{RL}/F_{\max}^{RL}$ 表示该作业的可用机器集中机器的最小负载和最大负载的比值; L_{\min}/L_{\max} 表示该作业在所有

可用机器上的最短可能加工时间和最长可能加工时间的比值。此外顶层 Q 学习还有一种虚状态, 即对于某道作业来说, 其可用机器只有 1 种。

2.3.3 奖惩函数设计

DA 着眼于提高系统的整体性能, 因此本文提出了 2 种特性指标衡量 Agent 的动作奖惩, 以便综合考虑所有机器上任务的拖期完成情况以及所有机器的负载平衡情况。

1) 平均作业拖期期望 F^{EMT} 。即 DA 动作后, 所有机器都按照当前最优原则, 将缓冲区内的作业安排加工后所有作业的平均拖期期望, 如式(6)所示。

$$F^{EMT} = \frac{1}{m} \cdot \sum_{i=1}^m \left(\frac{1}{N_{b,i}} \sum_{j=1}^{N_{b,i}} C_j^T \cdot \max(T_j^A - T_j^D, 0) \right). \quad (6)$$

2) 负载平衡因子 F^{RB} 。即所有机器的利用率的方差, 如式(7)所示。

$$F^{RB} = \sqrt{\frac{1}{m} \cdot \sum_{i=1}^m (F_i^{UR} - F^{MRU})^2}. \quad (7)$$

式中 F^{MRU} 表示机器利用率均值。

基于以上 2 种指标, 设定 DA 的奖惩值如表 6 所示。以奖励类别 1 为例, 如果 DA 的动作使负载平衡因子变大, 并且平均作业拖期期望减小, 那么 DA 收到的立即奖惩为 1。

表 6 顶层 Q 学习奖惩函数

Table 6 Reward function of top level Q-learning

奖励类别	判别标准	奖 惩
1	$F^{RB} \nearrow, F^{EMT} \searrow$	$r = 1$
2	$F^{RB} \nearrow, F^{EMT} \nearrow$	$r = -2$
7	$F^{RB} \searrow, F^{EMT} \searrow$	$r = 2$
8	$F^{RB} \searrow, F^{EMT} \nearrow$	$r = -1$

3 仿真实验

为了对提出的双层 Q 学习算法进行验证, 构建了如下动态调度仿真模型。系统由 M 台不同机器组成, 从 0 时刻开始, 作业随机进入系统缓冲区, 每完成 N 个作业称为 1 个 episode。相邻 2 个作业到达系统的时间间隔服从均值为 I 的指数分布, 每个作业包含的工序数目从 $[a_1, a_2]$ 之间随机选取, 每道工序的加工时间从 $[b_1, b_2]$ 之间随机选取, 每道工序随机选择 $[c_1, c_2]$ 种可用加工设备, 作业的拖期惩罚金 C_i^T 服从均匀分布 $U(d_1, d_2)$ 。每个作业的交货期为

$$T_i^D = T_i^A + f_1 l_i. \tag{8}$$

式中: f_1 为交货因子,服从均匀分布 $U(e_1, e_2)$. 作业的最早开始时间为

$$T_i^S = T_i^A + f_2 l_i. \tag{9}$$

式中: f_2 为开始因子,服从均匀分布 $U(f_1, f_2)$.

表7 仿真实验参数

Table 7 Parameters of simulation experiment

参 数	取 值			
	案例1	案例2	案例3	案例4
意义	HL/TD	HL/LD	LL/TD	LL/LD
M	10	10	10	10
N	1 000	1 000	1 000	1 000
I	11	11	12	12
$[a_1, a_2]$	[6,10]	[6,10]	[6,10]	[6,10]
$[b_1, b_2]$	[10,20]	[10,20]	[10,20]	[10,20]
$[c_1, c_2]$	[1,4]	[1,4]	[1,4]	[1,4]
$[d_1, d_2]$	[1,3]	[1,3]	[1,3]	[1,3]
$[e_1, e_2]$	[2,2.5]	[2,3.5]	[2,2.5]	[2,3.5]
$[f_1, f_2]$	[0,0.5]	[0,0.5]	[0,0.5]	[0,0.5]
α	0.1	0.1	0.1	0.1
γ	0.9	0.9	0.9	0.9
ε	0.1	0.1	0.1	0.1

仿真实验中采用了4个案例以便比较全面地反映实际生产中常见的动态调度情形,各参数取值如表7所示.表中HL的含义是高负载,即任务到来的间隔较小,LL的含义是轻负载,即任务到来的间隔较大,TD的含义是交货期较紧,LD的含义是交货期较松.为了避免随机因素影响,对每个案例重复仿真实验10次,每次对1000个episode进行仿真,并将不同调度策略下得到的作业平均拖期惩罚值列于表8之中.

表8 不同调度方法的平均拖期惩罚比较

Table 8 Average tardiness penalty comparison of different scheduling methods

机器选择/调度算法	平均拖期惩罚			
	案例1	案例2	案例3	案例4
随机选择/SPT规则	94.1	78.4	82.2	68.9
随机选择/EDD规则	103.7	75.3	70.8	59.7
随机选择/FIFO规则	112.5	86.4	69.3	62.7
随机选择/Q学习调度	92.3	74.9	68.3	53.6
Q学习选择/Q学习调度	84.4	70.3	64.1	48.1

可以看出,采用Q学习机制后,调度效果明显强于采用某一固定的调度规则;而加入了面向机器

分配的顶层Q学习后,系统性能得到了进一步的改善.究其原因,是因为顶层Q学习的引入使得分配到各加工机器的作业更合理、平衡,从而避免了不同机器的忙闲不均.为了说明这一现象,以案例1为例对单层Q学习和双层Q学习在调度过程中的机器负载分布情况进行比较.即对1000个episode进行仿真,每10个episode对机器利用率的方差进行计算,结果如图4所示.可以看出,双层Q学习算法使得机器利用率的方差减小,也就是说各机器的利用率更加接近、更加均衡.

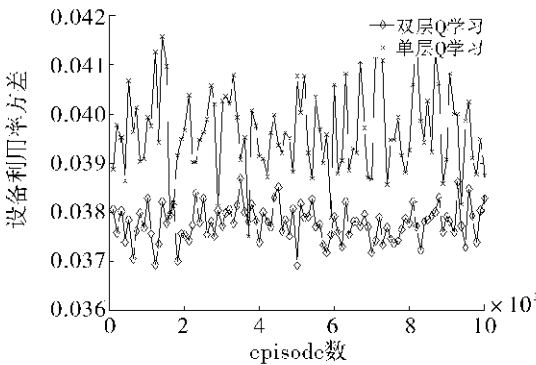


图4 单层Q学习和双层Q学习的机器利用率对比

Fig. 4 Machine utilization comparison of single-layer Q-learning and two-layer Q-learning

4 结束语

在单机动态调度问题中,Q学习已被证明可以根据缓冲区状态自适应地选择合适的调度规则;但是在多机动态调度环境下,不但需要关注单个机器的调度性能,还需要从全局角度考虑整体性能,否则难以取得良好的调度效果.因此构建了包含机器Agent和分配Agent的多Agent动态调度模型,并提出了相应的两层Q学习算法.机器Agent利用底层Q学习优化局部调度性能,分配Agent利用顶层Q学习优化任务分配,从而达到优化整体调度性能的目的.仿真试验表明,提出的两层Q学习算法能够明显地改善多机动态环境下的调度效果.

将Q学习应用于动态调度还有许多问题值得进一步的研究和深入探讨,例如更合理状态特征的提取、更有效的动作集、信息不完全情况下的强化学习,以及多Agent的合作强化学习等等.

参考文献:

[1]严浙平,李 锋,黄宇峰.多智能体Q学习在多AUV协

- 调中的应用研究[J]. 应用科技, 2008, 35(1): 57-60.
- YAN Zheping, LI Feng, HUANG Yufeng. Research on application of multi-agent Q-learning in multiAUV coordination[J]. Applied Science and Technology, 2008, 35(1): 57-60.
- [2] 潘燕春, 冯允成, 周 泓, 等. 强化学习和仿真相结合的车间作业排序系统[J]. 控制与决策, 2007, 22(6): 675-679.
- PAN Yanchun, FENG Yuncheng, ZHOU Hong, et al. Reinforcement learning integrated with simulation for job-shop scheduling system[J]. Control and Decision, 2007, 22(6): 675-679.
- [3] AYDIN M E, OZTEMELE E. Dynamic job-shop scheduling using reinforcement learning agents[J]. Robotics and Autonomous Systems, 2000, 33(2/3): 169-178.
- [4] WANG Y C, USHER J M. Application of reinforcement learning for agent-based production scheduling[J]. Engineering Applications of Artificial Intelligence, 2005, 18(1): 73-82.
- [5] WANG Y C, USHER J M. Learning policies for single machine job dispatching[J]. Robotics and Computer Integrated Manufacturing, 2004, 20(6): 553-562.
- [6] 魏英姿, 赵明扬. 强化学习算法中启发式回报函数的设计及其收敛性分析[J]. 计算机科学, 2005, 32(3): 190-193.
- WEI Yingzi, ZHAO Mingyang. Design and convergence analysis of a heuristic reward function for reinforcement learning algorithms[J]. Computer Science, 2005, 32(3): 190-193.
- [7] 王世进, 孙 晟, 周炳海, 等. 基于 Q 学习的动态单机调度[J]. 上海交通大学学报, 2007, 41(8): 1227-1232.
- WANG Shijin, SUN Sheng, ZHOU Binghai, et al. Q-learning based dynamic single machine scheduling[J]. Journal of Shanghai Jiaotong University, 2007, 41(8): 1227-1232.
- [8] 杨宏兵, 严洪森. 知识化制造系统中动态调度的自适应

策略研究[J]. 控制与决策, 2007, 22(12): 1335-1340.

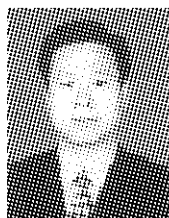
YANG Hongbing, YAN Hongsen. Adaptive strategy of dynamic scheduling in knowledgeable manufacturing system[J]. Control and Decision, 2007, 22(12): 1335-1340.

- [9] WATKINS C, DAYAN P. Technical note: Q-learning[J]. Machine Learning, 1992, 8(3/4): 279-292.

作者简介:



王国磊, 男, 1982 年生, 博士研究生. 主要研究方向为生产计划和车间调度等, 发表学术论文 10 余篇.



钟诗胜, 男, 1964 年生, 教授, 博士生导师. 哈尔滨工业大学威海分校副校长、中国机械工程学会机械设计分会理事、中国人工智能学会可拓专业委员会常务理事、中国工程图学学会应用图学专业委员会委员、全国工业自动化系

统与集成标准化技术委员会委员、国防科工委信息技术应用标准化技术委员会委员. 主要研究方向为数字化设计与制造、人工智能理论与应用、数控设备研发等. 国家 863/CIMS 重大应用示范工程项目——“HEC - CIMS II 工程”的副总设计师, 主持国家自然科学基金项目 2 项、国家 863 计划项目 2 项, 参与国家 863 计划项目 1 项、国家自然科学基金项目 1 项, 承担欧盟科技计划项目(英国、中国、西班牙联合承担)1 项, 多项省(部)级科技项目和企业横向项目. 曾获省部级科技进步二等奖 1 项、三等奖 2 项, 专利 1 个和国家自主知识产权登记软件 3 套, 被评为黑龙江省 CIMS 应用示范先进个人. 发表学术论文 140 余篇, 出版专著 1 部.



林 琳, 女, 1973 年生, 副教授, 硕士生导师. 主要研究方向为智能设计和产品数据管理等. 发表学术论文 20 余篇.