文章编号: 1001-0920(2013)02-0161-08

基于多 Agent 可互操作知识化制造动态自适应调度策略

汪浩祥, 严洪森

(东南大学 a. 自动化学院，b. 复杂工程系统测量与控制教育部重点实验室，南京 210096)

摘 要: 针对知识化制造系统生产环境的不确定性, 构建一个基于多 Agent 可互操作的知识化动态调度系统. 该系统对各种调度问题采用具有一系列问题特征的知识表示, 利用 Agent 技术构建基于问题的功能模块, 提出一种基于改进 Q-学习算法 (WSQ) 的自适应调度机制, 以此指导设备 Agent 在动态环境下的调度策略选择. 通过对其进行复杂性分析和仿真实验, 验证了该控制策略的有效性. 该系统具有自适应和自学习特征, 具有高度智能化和可互操作性.

关键词: 知识化制造系统；动态调度；多 Agent 技术；Q-学习

中图分类号: TH165 **文献标志码:** A

Interoperable dynamic adaptive scheduling strategy in knowledgeable manufacturing based on multi-agent

WANG Hao-xiang, YAN Hong-sen

(a. School of Automation，b. Key Laboratory of Measurement and Control of CSE，Ministry of Education，Southeast University，Nanjing 210096，China．Correspondent：WANG Hao-xiang，E-mail：whx39@hotmail.com)

Abstract: Aiming to the uncetainty of production enviroment in knowledgeable manufacturing system, an interoperable knowledgeable dynamic scheduling system based on multi-agent is built, in which a knowledge representation with a series of problem characteristics for various scheduling problems is adopted and problem-based function modules are constructed by using agent technology. An adaptive scheduling mechanism based on modified Q-learning agrorithm known as weighted subordination based Q-learning(WSQ) is proposed for guiding the equipment agent to select scheduling strategy in a dynamic enviroment. By the analysis of its complexity and simulation experiments, the results show the effectiveness of this strategy. The system is of adaptive and self-learning features, and high intelligence and interoperability.

Key words: knowledgeable manufacturing system; dynamic scheduling; multi-agent; Q-learning

0 引 言

知识化制造是文献 [1] 于 2001 年提出的新的制造理念. 该理念将一种先进制造模式看作一种先进制造知识, 通过 Agent 网与知识网 (KM) 之间一一对应的同构映射关系, 将已有的先进制造模式纳入知识化制造系统 (KMS)[2], 以满足各类制造企业的不同需求. 它以时间、质量、成本、服务和环境为主要目标, 具备自适应、自学习、自进化、自重构等特征, 是一种高智能制造系统. 近年来, 随着对 KMS 研究的逐步深入, 不断有新的研究成果[3-5]相继发表. 自适应是知识化制造系统的主要特征, 如何在动态多变的运行环境下, 实现具有不确定信息的知识化制造系统的动态自适应, 是知识化制造系统优化的关键技术.

实际生产过程大多具有大规模、带复杂约束和不确定等综合复杂性, 因此研究一种动态调度的自适应策略具有重要的意义. 动态调度区别于静态调度的两个最主要特征是调度的鲁棒性和对突发事件的反应能力[6], 所以交互式动态调度和在线调度应该是研究知识化制造系统自适应策略的主要研究方向. 目前, 很多学者进行了这方面的研究工作, 例如 Willy 等[7]针对不确定性项目调度问题分析了反应式调度、随机调度、模糊调度和鲁棒调度等; 杨宏兵等[5]利用 B-Q 学习算法, 构建了一种适用于 KMS 中动态调度问题的自适应调度控制策略; 包振强等[8]针对基于知识的调度系统在实际应用中存在的缺陷, 提出了一种基于知识的动态调度决策方法; 徐赐军等[9]针对产品开发

收稿日期: 2011-10-16；**修回日期:** 2012-01-03.

基金项目: 国家自然科学基金重点项目(60934008)；江苏省普通高校研究生科研创新计划项目(CXLX11−0118)；东南大学优秀博士论文基金项目(YBJJ1215).

作者简介: 汪浩祥(1979−), 男, 博士生, 从事知识化制造、复杂制造系统调度的研究；严洪森(1957−), 男, 教授, 博士生导师, 从事知识化制造、生产计划与调度等研究.

过程中由于活动变化导致原调度需重新调整的问题, 提出一种利用弹性资源特性进行动态调度决策的方法. 然而, 动态调度的现有研究主要集中在算法上, 难以达到预期效果, 面对现代制造的复杂要求, 需从系统的角度出发, 采用新的思路研究理想的调度系统模型、合理的调度系统结构以及能够集成经典调度方法的先进调度策略和方法.

考虑到多 Agent 系统用自主模块构成的分布式结构代替传统的集中式非自主性结构, 实际的调度执行主要通过多个代理协商来完成, 而不是完全的预先计划, 因此具有良好的灵活性和可互操作性, 非常适合处理动态调度问题[10]. 本文将设备封装成 Agent, 采用基于移动 Agent 与强化学习相结合的协商机制, 建立了基于多 Agent 的动态调度模型. 同时, 针对现有 Q 学习策略在动态调度应用中存在状态空间维数过大, 收敛速度慢的问题, 通过状态聚类方法减少维数又存在搜索精度方面的不足, 本文通过在 Q 值更新中加入最大模糊收益因子, 采用状态隶属度作为权系数对聚类状态的 Q 值进行加权迭代, 在每次更新中同时对多个聚类状态-动作对的 Q 值进行更新, 提出了一种基于聚类状态隶属度加权的 Q-学习机制 (WSQ), 以此指导设备 Agent 在动态环境下的调度策略选择. 该调度控制策略通过状态隶属度加权迭代的方式, 减少了由于状态聚类而造成的聚类状态与系统真实状态误差, 在减少系统状态空间维数的情况下, 使得对聚类状态的学习更接近于系统的真实状态, 提高了算法搜索的精度和遍历速度.

为此, 首先借鉴先全局再局部的设计思想, 利用 Agent 技术构建一个可互操作的知识化制造多 Agent 动态调度系统.

1 基于多 Agent 可互操作的知识化制造动态调度系统

Agent 智能的实现过程, 实际上是对知识的处理过程. 知识描述对象和状况的关系主要包括 3 个方面: 知识的获取, 知识的表示和知识的运用[11]. 因此, 知识表示是 Agent 处理信息的一个重要任务, 其表示方法的不同, 对于知识的利用程度和效率都存在很大的差异.

1.1 以生产运作管理问题为中心的知识化动态调度系统的知识表示

知识化制造的核心是生产运作管理中存在的知识, 提炼出这些知识并分析其特征是实现知识化制造的基础. 综合考虑知识化制造系统自身及所处环境的动态多变性和不确定性, 采用经验和仿真分析的方法, 在知识化制造系统调度知识库内, 可以将调度知识分成基于问题类型的知识单元和基于结论类型的知识单元[12].

基于问题类型的知识单元用一个三元组 $\mathrm{KU}_1 = (O, P, S)$ 进行描述, O 表示对象 (即制造系统), P 表示生产运作管理问题, S 表示解决方案. 同理, 一个基于结论类型的知识单元可由三元组 $\mathrm{KU}_2 = (O, \mathrm{KN}, C)$ 进行描述, O 表示对象, KN 表示已知参数集合, C 表示结论 (即制造系统的性质), 其中对象的定义与问题型知识单元中对象的定义相同. 结论可进一步描述为二元组 $C = (\mathrm{PF}, \mathrm{PR}_1)$, $C = (\mathrm{PA}, \mathrm{PR}_2)$, 或 $C = (\mathrm{PF}, \mathrm{PA}, \mathrm{PR}_3)$, PF 为性能集合, PA 为参数集合, PR_1 为不同性能之间性质的集合, PR_2 为不同参数之间性质的集合, PR_3 为性能与参数之间性质的集合, 其中性能与参数之间的性质实际上刻画了参数对性能的影响.

知识单元可以用结构化的语言表示. 例如: 知识单元 (对于由多台机器构成, 允许故障、缓冲区容量有限的串行生产线, 可采用近似分解数值算法分析其平均生产率和平均在制品水平) 可用图 1 表示.

FOR a tandem production line	Object
WITH	Knowns
{machine_quantity = n	
production_capacity = $(\mu_1,\cdots,\mu_n)$;	
failure_rate = $(p_1,\cdots,p_n)$;	
repair_rate = $(r_1,\cdots,r_n)$;	
buffer_capacity = $(N_1,\cdots,N_{n-1})$};	
TO_EVALUATE	Unknowns
{average production rate;	
average WIP level;}	
BY_USING	Solution
{The approximate decomposition	
algorithm}	

图 1 用结构化语言表示的知识单元

图 1 给出了基于以上问题的知识单元基本结构, 左列给出了知识单元的对象 (生产系统)、已知量 (包括参数和性能)、未知量 (包括参数和性能)和解决方案, 右列是存储这些信息的具体单元.

知识单元不是相互独立的, 而是相互联系的. 例如缓冲区容量分配优化问题以生产率评估问题为子问题: 不同的生产优化问题针对的可能是同一个制造系统, 需共享该制造系统的数据, 相互联系的知识单元共同构成了生产管理知识系统.

1.2 基于多 Agent 可互操作知识化制造动态调度系统建模

以生产运作管理问题为中心, 对调度问题知识进行表示和组织, 将知识化制造系统调度知识组织成一个知识系统, 可以有效地利用已有的知识来解决新的生产调度问题. 为此, 应设计能够重用已知知识的问

题求解机制, 而问题求解机制的实现又依赖于其运作的技术体系.

为了解决上述问题, 本文在信息整合的基础上, 应用基于 J2EE 和 JADE 的多 Agent 应用系统开发技术来实现知识化制造系统的自适应调度决策优化 (见图 2), 设计了基于多 Agent 的可互操作知识化制造动态调度系统模型. 系统从解决不同的生产调度问题出发, 通过各 Agent 组之间的交互操作, 利用与共同的对象模型之间的数据接口实现相关数据的传递, 从而解决不同的生产调度问题, 实现模型之间的互操作[12].

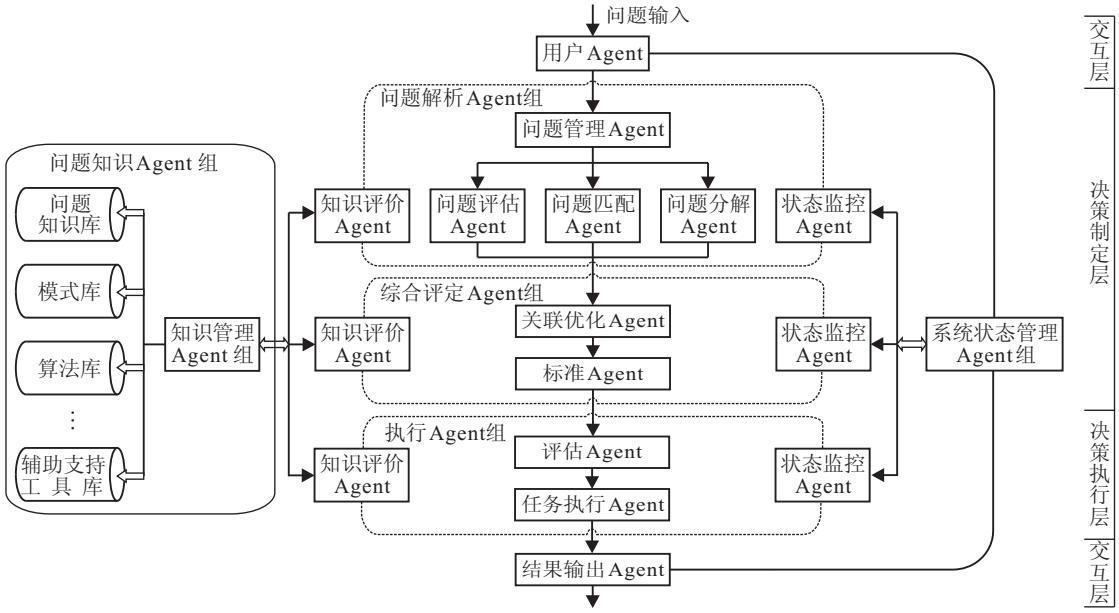

图 2　基于多 Agent 可互操作知识化制造动态调度系统结构模型

1.2.1　系统组成和功能层次

系统由 6 个 Agent 组承担: 用户 Agent 组、问题解析 Agent 组、综合评定 Agent 组、执行 Agent 组、问题知识 Agent 组和系统状态监控 Agent 组. 一个 Agent 组是由多个 Agent 围绕一个任务构成一个群体, 根据具体任务的不同设计出完成各种特定任务的专门结构. 它们是多 Agent 调度系统功能实现的核心部分, 各 Agent 相互作用关系及系统求解过程如图 2 所示.

从功能实现的角度看, 系统又可以分为交互层、决策制定层和决策执行层.

1.2.2　系统特点分析

与现有基于信息整合的调度系统一般结构相比, 本系统具有如下特点:

1) 实现的整合目标不同. 本调度系统生成的整合任务综合反映了一定时间内用户问题的所有要求, 包括较高层次的要求. 系统所建立的全局数据模型能够体现出用户的各方面要求, 而一般的此类系统方案则仅保证用户能够及时准确地查询所要求的数据, 这是信息整合中较低层次的要求.

2) 动态性. 主要体现在可以根据用户调度问题的不同要求, 制定不同的整合方案, 而在目前许多整合方案中, 任何用户都是基于一种方法进行整合. 此外, 根据用户要求的变化, 该系统本身具有适应性, 可以根据环境变化作出相应的调整.

3) 采用自适应型 Agent, 能够实现根据环境变化和用户的要求进行动态调整, 从而使得该动态系统自身也是一个自反馈型的动态系统.

4) 该系统具有学习能力. 在目前的一些整合方案中, 辅助管理 Agent 完全处于提供辅助支持的从属地位. 然而, 在本系统中扩充了辅助管理 Agent 的功能, 建立了问题知识 Agent 组, 它除对整合全过程的支持之外, 还可以分类存储整合过程中积累的新知识. 同时, 每个 Agent 组中的知识评价 Agent 本身便具有筛选和评价新知识的能力.

为实现各 Agent 之间的通信, 本文基于昆虫感应的协商策略, 由一类特殊 Agent ——“Mobile Agent”来协调系统内代表其他物理设备或逻辑功能的 Agent 之间的关系, 采用 Mobile Agent 与强化学习相结合的协商机制, 通过 Agent 之间的交互来实现实际生产车间的动态调度.

需要指出的是, 限于篇幅, 本文只给出了系统的结构模型, 没有给出各 Agent 之间的具体协商过程, Mobile Agent 在调度中与各 Agent 交互协调的具体过程可参见文献 [13], 在此不再详述.

2　问题描述

在基于可互操作的多 Agent 知识化制造动态调度系统工作过程中, 有一个重要环节是当系统遇到一个新的调度问题或在某时刻发生随机事件而引起调

度环境变化时, 由任务管理Agent接收任务, 通过问题库查找问题知识单元Agent的功能, 对任务所涉及的能力和问题知识单元Agent所具有的能力进行评价和选择, 从而完成问题的调度和分配, 并通过Agent之间的交互来实现实际生产车间的动态调度, 最后到达执行Agent进行实施, 得到用户满意的最优方案.

为便于描述, 本文给出如下符号定义: 某车间或工厂的加工设备集合为$V=\{M_1,M_2,\cdots,M_M\}$; 生产任务或作业集表示为集合$J=\{J_1,J_2,\cdots,J_N\}$; 每个作业由多道工序组成, O_{ij}代表作业J_i的第j道工序加工时间; 一台设备在同一时间内只能完成一道作业, 且同一作业的相邻两道工序不能由同一个加工设备完成, 第i道作业J_i的交货期为D_i, 实际完工时间为C_i. 按第1.1节的知识表示方法, 每个调度问题可用调度知识表示为

$$P=\{M,N,\lambda,n_1,n_2,u_{t1},u_{t2},u_{f1},u_{f2},\varepsilon,\gamma\}.$$

其中: M表示加工Agent数; N为任务作业数; $\lambda,n_1,n_2,u_{t1},u_{t2},u_{f1},u_{f2},\varepsilon,\gamma$分别为加工过程中的参数, 具体定义详见本文第4节; 调度目标为最小化作业平均拖期

$$\text{Object}=\min\Big\{\sum_{i=1}^{N}\max(C_i-D_i,0)/N\Big\}.$$

3 基于改进的Q学习算法WSQ的系统自适应调度策略

为解决上述问题, 本文给出一种能够自适应环境变化的动态调度控制策略: 基于改进的Q学习算法WSQ, 用于解决不同的动态调度问题.

3.1 WSQ算法

Q学习是一种典型的强化学习方法, 具有自适应、在线学习、试错以及自我选择的特点, 特别适于解决系统扰动较多的动态调度问题, 已在控制、决策及规划中得到广泛的应用[14]. 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函数来获得最优控制策略, 很多学者都做了这方面的尝试. 例如Aydin等[15]利用Q-III算法训练智能体动态选择调度规则, 王世进等[16]将强化学习应用于动态单机调度研究, Stefan[17]将强化学习用于flow-shop调度问题. 其后, 杨宏兵[5]和王国磊等[18]也分别研究了利用Q学习算法实时选择调度规则.

传统Q学习中通常每次学习只对一个Q值进行更新, 文献[18]虽然应用了一次迭代多个更新的方式, 但在其更新速度和精度方面仍有不足, 当动态调度问题的状态空间过大时, Q学习的一个固有的收敛速度慢的问题始终难以解决. 本文结合知识化制造Agent的高智能特征, 在现有Q学习算法的基础上提出了WSQ学习算法. 该算法首先利用顺序聚类方法(BSAS)[19]降低状态空间的复杂性, 用聚类状态-动作值表代替瞬时状态-动作值表, 并将系统状态模糊化, 利用状态隶属度作为权系数对Q值进行加权迭代更新, 从而提升了迭代过程中的值函数遍历速度和精度. 在本调度系统中, 系统通过各Agent组之间的交互操作实现系统与环境的交互, 进而不断地改进策略, 其目标是得到最优评估函数值Q^*, 即获得对应的最优控制策略.

3.2 状态空间选取

状态空间的划分是系统合理选择调度规则的基础. 然而, 在调度问题不断变化的动态调度中, 完整的系统状态是连续的, 且往往可由十几个甚至几十个状态特征刻画. 为避免由此导致的Q学习"维数灾难"问题, 经过大量的仿真实验, 本文选取4种对调度规则性能影响较大的状态特征指标, 并结合本文第1.1节的知识表示方法, 将t时刻系统状态用一个四元组进行刻画, 即$S_t=(L,F,U,T)$, L表示系统相对机器负载, F为平均交货因子, U为系统利用率, T表示平均松弛时间, 然后采用聚类方法划分状态空间. 该方法可以大大降低状态空间的复杂度. 下面对4个状态特征进行定义.

1) 设备负载率L. 令t时刻制造单元中各缓冲区中的作业剩余加工时间为ω, 将最大剩余加工时间与平均剩余加工时间的比值$L=\omega_{\max}/\varpi$称为相对机器负载.

2) 平均交货因子F. 反映作业交货期的松紧程度, 用式$F=\dfrac{\sum f_i}{N}$表示. 其中: f_i为知识化制造单元中第i个作业件的交货因子, 反映第i个作业交货期的松紧程度; N为系统中的作业总数.

3) 设备利用率U. 含义是知识化制造单元中当前非空闲加工Agent数与总的加工Agent数之比, 即$U=M_u/M$. 其中: M_u代表当前非空闲加工Agent数, M为总的加工Agent数.

4) 平均松弛时间T. 第j个作业的松弛时间表示为$T_j=D_j-t-\sum_{p=k_p}^{k_j}O_{jp}$. 其中: t为当前时刻, O_{jp}为第j个作业的工序p所需加工时间(若工序p正在被加工, 则O_{jp}为该工序的剩余加工时间), k_p为作业正在被加工或等待加工的工序数, k_j为作业j的工序总数. 则有

$$T=\Big(\sum_{j=1}^{N}T_j\Big)\Big/N.$$

基于以上4种特性指标, 结合基本顺序算法方案(BSAS)[19]对系统状态进行聚类. 聚类前, 为平衡各特

征在聚类中的作用, 须对上述状态特征值进行标准化预处理, 为保持特征的原有语义, 本文采用比例因子法.

为方便描述, 令对系统状态进行聚类后得到 K 个聚类, 则将第 x 个聚类中所有系统状态的中心称为聚类状态 $C_x, x=1,2,\cdots,K$. 由于聚类状态在系统状态空间中只占 K 个孤立点, t 时刻系统状态往往与聚类状态存在一定的差别. 为了描述这种差别, 本文给出如下状态隶属度的概念.

定义 1　令 t 时刻系统状态 S_t 与聚类状态 C_x 的距离为 d_{tx}, 则称

$$\rho_{C_x}(S_t)=1-\frac{d_{tx}}{\max\limits_{1\leqslant x\leqslant K}(d_{tx})} \tag{1}$$

为状态 S_t 对聚类状态 C_x 的隶属度. 式 (1) 中

$$d_{tx}\in[0,\max_{1\leqslant x\leqslant K}(d_{tx})],$$

定义为状态 S_t 与第 x 个聚类状态 C_x 的 Euclid 距离, 即 $d_{tx}=\Big(\sum\limits_{i=1}^{q}(S_{ti}-C_{xi})^2\Big)^{1/2}$, 其中 q 为聚类状态向量 C_x 的维数, 则状态 S_t 对所有聚类状态的隶属度向量可以表示为 $\rho_C(S_t)=(\rho_{C_1}(S_t),\rho_{C_2}(S_t),\cdots,\rho_{C_x}(S_t),\cdots,\rho_{C_K}(S_t))$.

根据上述状态隶属度函数定义, 由 $d_{tx}\in[0,\max\limits_{1\leqslant x\leqslant K}(d_{tx})]$ 可得 $0\leqslant\rho_{C_x}(S_t)\leqslant 1$, 且状态 S_t 与聚类状态 C_x 的距离 d_{tx} 越小, $\rho_{C_x}(S_t)$ 的值越接近于 1; 反之, 状态 S_t 与聚类状态 C_x 的距离越大, $\rho_{C_x}(S_t)$ 的值越接近于 0. 特别地, 当 $d_{tx}=\max\limits_{1\leqslant x\leqslant K}(d_{tx})$ 时, 状态 S_t对聚类状态 C_x 的隶属度为 0, 此时认为状态 S_t 与聚类状态 C_x 不相似. 这样, 通过隶属度便可以充分反映系统状态与聚类状态之间的近似程度.

定义 2　令 S_t^c 为当前系统状态 S_t 对应隶属度最大所在的聚类状态, 即满足

$$\forall S_t^c\in C_x,\ S_t^c=\arg\max_{1\leqslant x\leqslant K}\rho_{C_x}(S_t),$$

则称 S_t^c 为当前状态 S_t 对应的聚类状态. 同理, S_{t+1}^c 表示下一时刻的聚类状态, 以下同.

定义 3　给定系统当前状态 S_t 对于各聚类状态的隶属度为 $\rho_{C_x}(S_t)$, 执行动作 a_t 后得到下一个状态 S_{t+1} 对于各聚类状态的隶属度为 $\rho_{C_x}(S_{t+1})$, $x=1,2,\cdots,K$, 与状态 S_{t+1} 对应的聚类状态为 S_{t+1}^c, 各聚类状态-动作值函数为 $Q(C_x,a)$, $\forall a\in A$, A 为系统动作集, 则称所有聚类状态下的最大收益值的加权和

$$\hat{Q}^{S_{t+1}}=\frac{\sum\limits_{x=1}^{K}(\rho_{C_x}(S_{t+1})\cdot\max\limits_{a\in A}(Q(C_x,a)))}{\sum\limits_{x=1}^{K}\rho_{C_x}(S_{t+1})} \tag{2}$$

为状态 S_{t+1} 的最大模糊收益, 权值为归一化的状态 S_{t+1} 对各个聚类状态的隶属度.

3.3　状态隶属度加权的 Q 值更新策略

针对 Q 学习要求 Agent 多次遍历状态-动作对值才能使估计 Q 值收敛于真实 Q 值这一特点, 利用系统状态监控 Agent 对外界状态判断的不确定性, 在一次迭代中对多个值函数进行更新, 以期尽快增加各种状态-动作对值的遍历次数, 提高搜索的精度和遍历速度. 基于定义 1 和定义 3 对状态隶属度和最大模糊收益的定义, 在标准 Q 学习迭代方程中增加一项最大模糊收益因子, 在每次迭代中同时对最大将来回报和最大模糊收益进行更新, 提出一种基于瞬时状态对聚类状态隶属度加权的 Q 值更新迭代策略, 如下式所示:

$$\begin{aligned}&Q_n(S_t^c,a_t)=\\&(1-\alpha_n)Q_{n-1}(S_t^c,a_t)+\\&\alpha_n\{r_{t+1}+\gamma\max_{b\in A}[\rho_{S_{t+1}^c}(S_{t+1})Q_{n-1}(S_{t+1}^c,b)+\\&(1-\rho_{S_{t+1}^c}(S_{t+1}))\hat{Q}_{n-1}^{S_{t+1}}]\}.\end{aligned} \tag{3}$$

其中: $Q_n(S_t^c,a_t)$ 为当前聚类状态 S_t^c 第 n 次循环时的 Q 值; $Q_{n-1}(S_t^c,a_t)$ 为第 $n-1$ 次循环的更新值; $\hat{Q}_{n-1}^{S_{t+1}}$ 为第 $n-1$ 次循环时状态 S_{t+1} 的最大模糊收益; r_{t+1} 为即时回报值; γ 为对延迟回报的折扣因子且 $0\leqslant\gamma<1$, $\rho_{S_{t+1}^c}(S_{t+1})(0\leqslant\rho_{S_{t+1}^c}(S_{t+1})<1)$ 为状态 S_{t+1} 对对当前聚类状态 S_{t+1}^c 的隶属度; α_n 为步长参数, 可由下式得到:

$$\alpha_n(S_t^c,a_t)=\frac{\phi_\alpha}{1+V_{S_n(S_t^c,a_t)}}. \tag{4}$$

式中: ϕ_α 为步长参数的非负权系数变量; $V_{S_n(S_t^c,a_t)}$ 为 n 次循环中, 状态-动作对 (S_t^c,a_t) 被访问的总次数; 步长参数 α_n 会随着 (S_t^c,a_t) 被访问次数的增加而减小. 动作搜索策略采用 ε 贪婪算法, 当从知识化制造单元的初始状态-动作对 (S_{t_0},a_{t_0}) 开始, 每步都采用 ε-greedy 法选取动作, 即以概率 $(1-\varepsilon)$ 选择具有最大状态-动作对函数值 $\max\limits_{a_t\in A}(Q(S_t^c,a_t)$ 的动作 a_t, 以概率 ε 随机选取动作集 A 中其他动作时, 则可得到最大的折算累积回报期望值, 即最优评估函数值 Q^*.

3.4　奖惩函数设计

奖惩函数设计的好坏决定了 Agent 能否快速从所选规则的效果中学会在不同环境中应作出何种选择. 文献 [20] 指出, 奖惩函数的设计应该对应系统的调度目标. 因为本文的目标函数是最小化平均拖期, 而 WSQ 学习算法收敛于最大值, 于是对 WSQ 学习算法中的立即回报值 r 设定如下:

$$r=\begin{cases}\Big(\sum\limits_{j=1}^{l}T_j\Big)\Big/(\overline{F}/\Phi), & \text{缓冲区工件发生拖期;}\\ 1, & \text{缓冲区工件不拖期.}\end{cases} \tag{5}$$

其中: $\sum_{j=1}^{l} T_j$ 为所有缓冲区中拖期工件的总松弛时间, T_j 为拖期工件 j 的松弛时间, l 为缓冲区中拖期工件数目; $\overline{F}$ 为缓冲区中(拖期)工件的平均交货因子; Φ 为一定常数, 并有 $\Phi > \max\left(\sum_{j=1}^{l} T_j \Big/ \overline{F}\right)$, 计算时除以 Φ 用于对 r 值进行数据归一化处理.

3.5 算法步骤

根据上述分析, WSQ学习算法的具体实现步骤如下:

Step 1: 置最大聚类数为 K, 问题解析 Agent 组采用基本顺序算法方案(BSAS)对系统状态进行聚类, 得到 K 个聚类状态 C_x, $x = 1, 2, \cdots, K$, 并将聚类状态存储到问题知识 Agent 组.

Step 2: 综合评定 Agent 组对所有聚类状态-动作对 $(C_x, a)(a \in A)$ 的 Q 值进行初始化, 并存储于问题知识 Agent 组知识库中, 记为 $Q_0(C_x, a)$; 置循环次数 $n = 1$, 在系统运行初始时刻 t_0, 执行 Agent 组从问题知识 Agent 组动作集中任选动作 a_{t_0} 进行调度.

Step 3: 由状态监控 Agent 组检测到当前系统状态 S_t, 反馈到综合评定 Agent 组, 综合评定 Agent 组调用问题知识 Agent 组知识库计算 S_t 对于各个聚类状态的隶属度向量 $\rho_C(s_t)$.

Step 4: 由综合评定 Agent 组按定义 2 求取当前状态 S_t 对应的聚类状态 S_t^c, 执行 Agent 组根据 ε 贪婪策略从问题知识 Agent 组动作集中选择当前具有最大回报值的动作 a_t (调度规则).

Step 5: 执行 Agent 组根据规则 a_t 从缓冲区选择作业进行加工, 通过式 (5) 计算立即回报值 r_{t+1}, 此时由状态监控 Agent 组观察并得到下一个状态 S_{t+1}, 由综合评定 Agent 组计算状态 S_{t+1} 对于各个聚类状态的隶属度向量 $\rho_C(s_{t+1})$ 和对应的聚类状态 S_{t+1}^c.

Step 6: 执行 Agent 组通过搜索问题知识 Agent 组知识库, 得到聚类状态 S_{t+1}^c 下的最大将来回报 $\max_{b \in A}(Q_{n-1}(S_{t+1}^c, b))$, 同时对多个聚类状态-动作对的 Q 值进行更新并存储, 并由综合评定 Agent 组计算得到状态 S_{t+1} 的最大模糊收益 $\hat{Q}_{n-1}^{S_{t+1}}$.

Step 7: 综合评定 Agent 组根据式 (3) 更新策略对值函数 $Q_n(S_t^c, a_t)$ 进行迭代更新并存储到问题知识 Agent 组知识库中, 然后置 $n = n + 1$.

Step 8: 置 $S_t = S_{t+1}$, 以更新状态, 并返回 Step 3.

Step 9: 重复 Step 3 ~ Step 8, 直到学习到所有状态-动作对的最优函数值 Q^*, 即最优控制策略.

Step 10: 结束.

需要指出的是: 本文算法中状态的聚类不是在线完成的, 从而避免了在线学习时由聚类状态改变而导致的收敛速度过慢的问题. 状态聚类的初始样本来自仿真器, 以后每次学习后都可以统计历史状态数据, 通过不断调整系统状态与对应聚类状态的隶属度阈值 Ω, 可使对应聚类状态逐渐逼近系统状态, 从而提高聚类的精度. 算法中 Step 3 ∽ Step 7 只对 $Q_n(S_t^c, a_t)$ 进行更新, 在每次更新中同时对第 $n-1$ 步中未知的聚类状态-动作对 Q 值进行更新, 从而实现对多个聚类状态-动作对的 Q 值更新. 更新一次就是一个循环, 当状态变换为另一个状态时, Step 8 用下一状态 S_{t+1} 代替当前状态 S_t, 开始对 $Q_n(S_{t+1}^c, a_t)$ 进行更新. 在整个调度过程中, 由 Mobile Agent 来协调系统内代表其他物理设备或逻辑功能的 Agent 之间的关系, 通过执行 Agent 组不断地与综合评定 Agent组、知识管理 Agent 组以及环境监控 Agent 组之间的交互操作, 利用与共同的对象模型之间的数据接口实现相关数据的传递, 实现模型之间的互操作, 最终获得最优控制策略的 Q^*.

3.6 算法复杂性分析

WSQ 算法的学习过程由情节和步骤组成, 其中步骤是指一个确定的状态及该状态下的动作执行和报酬获得, 情节是指从起始状态到目标状态的步骤序列. 为了与前面算法中的步骤相区分, 将 WSQ 学习中的步骤称为 RL 步骤.

由于每个 episode 中有多少 RL 步骤是不确定的, 这与算法的收敛速度有关, 这里仅考虑每个 RL 步骤中的计算复杂度. 在 WSQ 算法中, 每个 RL 步骤中要更新的聚类状态-动作对的 Q 值个数不会大于 $3K-1$ 个 (系统的聚类状态-动作对数目为 $3K$), 因此计算复杂度不大于 $O(3K-1)$, 为计算机可接受.

4 仿真实验

为了验证本文构建的多 Agent 调度模型及基于 WSQ 学习的系统自适应调度策略的有效性和实用性, 针对企业生产的复杂性以及生产环境和市场需求的不确定性易导致生产计划变动频繁、产生各种生产调度问题的情况, 本文模仿其调度环境, 设计了一个自适应动态调度仿真模型, 用于测试和分析. 系统由 M 台不同设备 Agent 组成, 从初始时刻开始, 作业随机进入系统, 相邻两个作业到达系统的时间间隔服从负指数分布, 平均到达率为 λ, 每个作业包含的工序数目是从 $[n_1, n_2]$ 之间随机选取的整数, 每道工序的加工时间服从均匀分布 $[u_{t1}, u_{t2}]$. 当进入作业的数目达到 N 后, 仿真停止. 调度规则库中调度规则选用最早交货期优先 EDD, 最短加工时间优先 SPT 和最小松弛时间优先 MST 共 3 个常用规则. 如果时间 t 有作

业到达或者需求发生改变, 则系统监控 Agent 检测当前系统状态, 获取系统的动态调度知识, 对调度知识库中的知识进行更新. 执行 Agent 将根据检测到的系统状态读取调度问题知识 Agent 组中对应的调度知识, 基于 WSQ 学习算法动态选取相关的调度规则对作业进行调度, 从而保证系统对不同调度问题的动态自适应.

在实验中, 第 i 个作业的交货期 D_i 设定为

$$D_i = A_{t_i} + f_i \sum_{q=1}^{k_i} O_{iq}.$$

其中: A_{t_i} 表示第 i 个作业到达系统时刻; O_{iq} 表示第 i 个作业的工序 q 所需加工时间; k_i 表示作业 i 的工序总数; f_i 为交货因子, 服从均匀分布 $f_i \sim U(d_{f1}, d_{f2})$.

本文给出 8 个基于上述模型的仿真案例, 代表调度系统面对的不同问题. 问题参数如表 1 所示.

表 1　不同调度问题主要参数

问题	M	N	λ	n_1	n_2	u_{t1}	u_{t2}	d_{f1}	d_{f2}
P_1	10	2 500	1/5	1	6	1	10	1	6
P_2	10	2 500	1/5.5	1	8	2	12	1	8
P_3	10	2 500	1/4.5	1	6	1	10	1	8
P_4	10	2 500	1/5.5	1	6	1	10	1	8
P_5	6	2 500	1/5	1	6	1	10	1	6
P_6	6	2 500	1/5.5	1	8	2	12	1	8
P_7	6	2 500	1/4.5	1	6	1	10	1	8
P_8	6	2 500	1/5.5	1	6	1	10	1	8

$M, N, \lambda, n_1, n_2, u_{t1}, u_{t2}, d_{f1}, d_{f2}$ 为如上述所描述的加工过程参数, 设置 WSQ 学习的贪婪策略系数为 $\varepsilon = 0.15$, 对延迟回报的折扣因子为 $\gamma = 0.7$. 其中: $P_1 \sim P_4$ 分别代表任务交货期较紧、任务结构差异较大(产品种类多)、任务到达间隔较小(市场需求较大)以及任务到达间隔较大(市场需求较小)的情况; 设置问题 $P_5 \sim P_8$ 为设备 Agent 减少为 6 台时对应情况的问题.

知识化制造 Agent 每处理完 2 500 个作业称为一个 episode, 为减小随机因素的影响, 对每个问题 (共对 200 个 episode) 进行仿真, 取平均拖期的总均值, 与文献 [5] 提出的 B-Q 策略和文献 [18] 提出的 CSMQ 策略进行比较, 结果如表 2 所示.

为进一步说明 WSQ 策略在动态调度环境中的优异性能, 以最复杂的调度问题 P_6 为例, 依次取每 20 个 episode 为一个批量计算其平均拖期的均值, 3 种策略的变化趋势如图 3 所示. 为显示 3 种算法的效率, 对 3 种策略的运行时间进行比较. 同样, 为减少随机因素, 从第 100 个 episode 开始取值. 3 种策略的时间变化趋势如图 4 所示.

由表 2 可以看出, 对于每种调度问题, 采用本文策略的调度结果较性能最好的策略提高了 1%～30%,

表 2　不同调度策略作业平均拖期比较

调度问题	200 个 episode 平均拖期的总均值			
	CSMQ 策略	B-Q 策略	本文策略	提高 / %
P_1	745.3	757.8	737.1	1.10
P_2	1 604.9	1 804.2	1 512.8	5.74
P_3	677.6	630.4	602.5	4.43
P_4	406.9	453.0	401.2	1.40
P_5	3 477.1	3 537.4	3 179.6	8.56
P_6	134 665	141 755	93 660	30.45
P_7	2 883.1	3 103.9	2 719.8	5.66
P_8	1 398.9	1 312.4	1 219.2	7.10

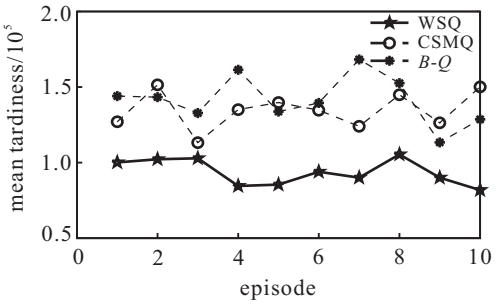

图 3　3 种策略平均拖期变化趋势

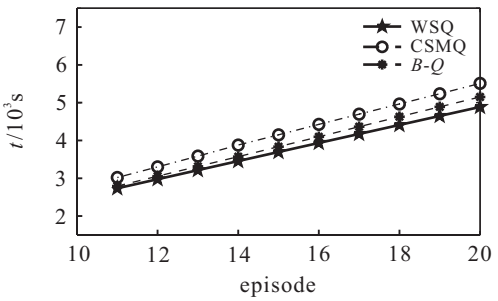

图 4　3 种策略求解效率比较

并随着调度问题复杂度的提高和设备 Agent 的减少显示出更好的效果, 突出了本文调度策略的优越性. 从图 3 可以看出, 本文 WSQ 策略较 CSMQ 策略和 B-Q 策略表现出了更好的稳定性, 每个批量的调度结果都令人满意. 从图 4 可以看出, WSQ 策略具有更好的求解效率, 说明本文策略通过状态隶属度加权迭代的方式, 减少了由状态聚类而造成的聚类状态与系统真实状态的误差, 在减少系统状态空间维数的情况下, 使得对聚类状态的学习更接近于系统的真实状态, 提高了算法搜索的精度和遍历速度.

5　结　论

本文针对知识化制造系统生产环境的不确定性, 构建了一种基于多 Agent 的可互操作知识化动态调度系统, 并基于该系统提出了一种基于改进的 Q 学习算法 WSQ 的动态自适应调度策略, 同时分析了该算法的复杂性. 该算法采用基于状态隶属度加权的 Q 值更新策略, 通过一次迭代中同时更新多个聚类状态-动作对的 Q 值的方式, 在 Q 值更新中加入最大模糊收益因子, 提高了搜索的精度和遍历速度. 数值仿真实

验表明, 在基于不同的调度问题情况下, 该调度控制策略明显优于文献[5]和文献[18]所提出的策略, 而且在调度问题复杂情况下能够显示出更好的性能, 提高了Q学习在构建知识化制造自适应调度策略中的实用效果.

参考文献(References)

[1] 严洪森, 刘飞. 知识化制造系统——新一代先进制造系统[J]. 计算机集成制造系统, 2001, 7(8): 7-11.
(Yan H S, Liu F. Knowledgeable manufacturing system—A new kind of advanced manufacturing system[J]. Computer Integrated Manufacturing Systems, 2001, 7(8): 7-11.)

[2] Yan H S. A new complicated knowledge representation approach based on knowledge meshes[J]. IEEE Trans on Knowledge and Data Engineering, 2006, 18(1): 47-62.

[3] 严洪森. 新的先进制造模式知识表示方法[J]. 机械工程学报, 2006, 42(10): 80-90.
(Yan H S. New approach to knowledge representation for advanced manufacturing modes[J]. Chinese J of Mechanical Engineering, 2006, 42(10): 80-90.)

[4] 薛朝改, 严洪森. 基于Agent网的知识网的自重构研究[J]. 计算机集成制造系统, 2003, 9(11): 995-1000.
(Xue C G, Yan H S. Self-reconfiguration of knowledge webs based on agent webs[J]. Computer Integrated Manufacturing Systems, 2003, 9(11): 995-1000.)

[5] 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 H B, Yan H S. Adaptive strategy of dynamic scheduling in knowledgeable manufacturing system[J]. Control and Decision, 2007, 22(12): 1335-1340.)

[6] 钱晓龙, 唐立新, 刘文新. 动态调度的研究方法综述[J]. 控制与决策, 2001, 16(2): 141-145.
(Qian X L, Tang L X, Liu W X. Dynamic scheduling: A survey of research methods[J]. Control and Decision, 2001, 16(2): 141-145.)

[7] Willy H, Role L. Project scheduling under uncertainty: Survey and research potentials[J]. European J of Operational Research, 2005, 165(2): 289-306.

[8] 包振强, 李长仪, 周鑫. 基于知识的动态调度决策机制研究[J]. 中国机械工程, 2006, 17(13): 1366-1370.
(Bao Z Q, Li C Y, Zhou X. A knowledge-based dynamic scheduling decision system[J]. China Mechanical Engineering, 2006, 17(13): 1366-1370.)

[9] 徐赐军, 李爱平, 刘雪梅. 弹性资源约束的动态调度决策[J]. 控制与决策, 2011, 26(3): 332-348.
(Xu C J, Li A P, Liu X M. Decision making for dynamic scheduling with flexible resource constraints[J]. Control and Decision, 2011, 26(3): 332-348.)

[10] 刘金琨, 尔联洁. 多智能体技术应用综述[J]. 控制与决策, 2001, 16(2): 133-140.
(Liu J K, Er L J. Overview of application of multi-agent technology[J]. Control and Decision, 2001, 16(2): 133-140.)

[11] 赵瑞清. 知识表示与推理[M]. 北京: 气象出版社, 1991: 1-9.
(Zhao R Q. Knowledge representation and reasoning[M]. Beijing: China Meteorological Press, 1991: 1-9.)

[12] Wang Z. Problem-oriented knowledge representing, organizing linebreak and inference for production operation and management[R]. Nanjing: School of Automation, Southeast University, 2010: 1-9.

[13] Bourenane M, Mellouk A, Benhamamouch D. State-dependent packet scheduling for QoS routing in a dynamically changing environment[J]. Telecommunication Systems, 2009, 42(3/4): 249-261.

[14] 陈宗海, 文锋. 基于复杂过程简化模型的DHP学习控制[J]. 控制与决策, 2006, 21(10): 1087-1091.
(Chen Z H, Wen F. Learning control of DHP method based on complex process simplified model[J]. Control and Decision, 2006, 21(10): 1087-1091.)

[15] Aydin M E, Öztemel E. Dynamic job-shop scheduling using reinforcement learning Agents[J]. Robotics and Autonomous Systems, 2000, 33(2): 169-178.

[16] 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, 2002, 36(3): 224-230.
(Wang S J, Sun S, Zhou B H. Q-learning based dynamic single machine scheduling[J]. J of Shanghai Jiaotong University, 2002, 36(3): 224-230.)

[17] Stefan P. Flow-shop scheduling based on reinforcement learning algorithm[J]. Production Systems and Information Engineering, 2003, 1(1): 83-90.

[18] 王国磊, 林琳, 钟诗胜. 基于聚类状态隶属度的动态调度Q-学习[J]. 高技术通讯, 2009, 19(4): 428-433.
(Wang G L, Lin L, Zhong S S. Clustering state membership-based Q-learning for dynamic scheduling[J]. Chinese High Technology Letters, 2009, 19(4): 428-433.)

[19] Theodoridis S, Kout roumbas K. Pattern recognition[M]. 2nd ed. San Diego: Academic Press, 2003: 634-635.

[20] 魏英姿. 制造系统生产调度和机器人学习智能研究[D]. 沈阳: 中国科学院沈阳自动化研究所, 2004.
(Wei Y Z. The study of product scheduling and robot intelligent learning for manufacturing systems[D]. Shenyang: Shenyang Institude of Automation, Chinese Academy of Science, 2004.)