**基于多Agent的动态调度系统的设计和研究**

1. **绪论**
   1. **研究的背景和意义**

当今制造业正经历深层次的变革，生产模式正逐渐从从“小品种，大批量”到“多品种，小批量”的转变，这是由于科学技术的快速发展，人们生活水平有了极大的提高，人们的生活需求不再局限于基本的物质生活保障，而是追求商品的个性化。面对制造领域市场日益激烈的竞争，制造企业必须满足多样化市场需求以及提升产品竞争力，才能获取更大的市场份额和产品利润。激烈的市场竞争要求制造企业具有能够适应持续波动的市场需求变化的能力以及响应制造执行过程中出现的生产扰动因素的能力。然而，传统的制造业在作业调度上主要以面向静态任务为主，一旦生产任务发生变化，通常需要在停工的前提下完成资源调整。而制造企业在持续变化的动态生产环境中，很难通过不断停产完成制造系统的动态调整，即使能够完成针对当前任务的系统规划，也无法快速响应实时的任务变化。此外，当实际生产中遭遇原料短缺、机器故障、质检不合格而返工等干扰时，就引起了生产运作的重调度问题，即对原有系统调度方案进行修改乃至重新生成。

车间作业调度是企业制造系统的关键模块，是体现制造系统生产管理先进性的核心。为了适应市场制造业日益普及的“小批量，多品种”的生产模式，我国企业生产车间普遍具有多种类产品混线生产的特点，系统作业调度需要考虑的因素较多，如产品种类的多样性、零件结构的多样性、产品工艺的多样性等等，再加上零件加工受工艺约束的工序排序和机器选择等柔性因素存在，使得制造系统的生产调度问题是随问题规模呈指数成倍增长的NP难类型。有很多研究者先后对该类问题进行了探索，得到了很多优化算法来专门解决这类问题，主要包括遗传算法、蚁群算法、模拟退火算法、粒子群优化算法等等。这些算法使得传统的作业调度的研究取得很大的进展，然而由于生产环境和生产需求的不确定性，实际作业车间调度问题比传统车间调度问题复杂很多，需要考虑随机性和不确定性等因素的影响。因此要解决现代制造中复杂的调度问题，需要研究新的、更实用的调度算法，同时为了能够快速响应快速的市场变化，我们还需要从系统的角度出发，研究理想的调度系统模型和合理的调度系统结构。

随着生产规模的增长以及生产模式的转变，现代制造业对动态调度系统显然有了更高的要求，除了需要具有传统调度系统应有的稳定性、可靠性、可维护性等特性外，还需要具有可适应性、敏捷性、可重构性等特性，目的是为了能够能够适应持续波动的市场需求变化的能力以及响应制造执行过程中出现的生产扰动因素。随着计算机技术的发展以及分布式人工智能的发展，多智能体系统（Multi-Agent System，MAS）为制造业领域的动态调度研究提供了新思路。MAS是高度模块化的系统，其下的各个Agent经良好的设计和封装，能够自主响应系统环境的信息并执行特定的行为。多个分布和并行工作的Agent通过协作完成某些特定的任务，其优点是速度快、可靠性高、可扩展性强、对不确定性数据和知识有较好的容错性。MAS提供了一种动态灵活、能快速响应市场需求变化和生产干扰事件的生产调度机制，有利于大大提高企业生产效率和资源利用率，进而增强企业的竞争能力。

本文把现代制造系统与MAS相结合，把制造系统中的制造资源结构、产品工艺结构、监控模块、调度算法等封装成对应的Agent，并对每个Agent设定能够响应的环境信息和相应的行为动作。当面临新的订单请求、订单变更、订单取消等需求波动因素以及制造执行过程中设备故障、原料短缺等生产扰动因素时，各Agent在各司其职的同时还需要相互协调合作，共同寻找出最优的调度方案，争取把干扰因素造成的影响降到最低。

* 1. **动态调度问题的研究现状**
     1. **作业车间调度问题的研究现状**

车间调度是指在一定时间内，为了满足特定的性能目标，对可用资源进行分配，对生产任务进行排序。车间调度问题具有计算复杂和和动态随机性的特点。例如加工过程中工件、原料、设备等因素会相互影响、相互作用、相互制约，随着问题规模增长所需的计算量会呈指数增长。再加上调度过程中随时可能会有新的加工任务插入或撤销、设备故障、原料短缺等不确定因素的存在，因此需要调度系统能够灵活高效地作出应对。

在上世纪七八十年代，研究人员的研究重心主要放在传统的车间调度问题，也就是面向静态的长期的调度任务，由于问题规模通常比较小，任务生产周期的时间跨度很长，因此响应的计算量也较小，通常采用的方法主要是精确算法，如枚举法，分支界定法等。随着计算机技术和人工智能的飞速发展，也随着对一些生物自然规律的认知加深，研究人员逐渐提出了更多行之有效的优化算法，这些算法克服了传统算法的局限性，更加适用于现代制造业的车间调度问题，这些算法包括，遗传算法、蚁群算法、粒子群算法、模拟退火算法、禁忌算法等。文献[1] 以实现面向制造执行过程的制造资源持续、动态优化配置为目标，以作业调度排产所体现的资源重构配置特点为出发点，研究生产调度与RMS相结合的理论方法和实现技术，推动RMS面向制造执行层次的转变，支持持续、动态的制造资源优化配置，最终开发了支持制造系统持续重构演变的实用工具。文献[2]克服了传统车间调度问题使用周期性重调度导致的效率低下问题，提出了一种基于多目标约束的局部遗传搜索算法，该算法能够保证车间调度的稳定性和高效性。文献[3] 在制造系统中综合使用了多Agent系统（MAS）以及分布式的遗传（DGA）算法，最终建立了MAS-DGA系统，系统在遗传算法中中还加入了协商机制，并与共生进化算法和合作协同进化算法进行了比较，证实了MAS-DGA在IPPS中灵活高效的特点。文献[4]在分析混合车间调度问题的特性后，在最小化最大完成时间的目标下，提出一种改进的遗传算法，即根据流水式作业和开放式作业的交叉进行切换算法的目标函数，从而获取更好的调度结果。文献[5]提出了一种改进的蚁群优化算法.通过节点集、有向弧集、无向弧集,构建了一种基于图的IPPS优化模型。以零件加工时间作为启发式信息，设计蚂蚁在各节点间转移概率。通过蚂蚁访问图中不同的节点，构建对应的调度方案。为避免陷入局部收敛，论文通过重启算法和重置各弧段信息素初值，动态更新各弧段信息素量，以获得全局最优解。文献[5]使用并行多种群遗传算法解决车间调度问题，作者对遗传算法进行了改进，使用特定的基因变异和交叉点来获取更好的调度结果。

* + 1. **基于多Agent调度系统的研究现状**

随着分布式人工智能的发展，Agent也应运而生。通常而言，具有自主性、社交性、反应性和能动性的硬件系统或软件系统被称为Agent。Agent能感知自身所处的环境，并对环境的改变作出实时的反应；能够自主控制自身的行为和内部状态，能够通过ACL（Agent Communication Language）与其他Agent进行交互。

近年来，随着对多Agent系统的进一步研究，表明拥有自主性、反应性和能动性的Agent之间彼此合作的分布式制造系统可以快速响应环境的变化，具有高效、灵活、稳定的良好特性。多Agent系统降低了集中控制的限制，提高了分散控制和并行处理的能力。这样，制造系统就可以通过分布在车间的资源和Agent之间的联合来实现系统的集成。近年来，很多研究人员把多Agent理论和多Agent技术应用在车间调度系统上，建立各式各样的基于多Agent的动态调度系统，取得了一定的成果。

文献[1]提出了一种基于现场总线的多Agent作业车间动态调度模型，首先由一个主Agent向其他辅助Agent发送调度任务，通过对回收的所有调度结果进行统一判断和协调，最终得到一个可行的调度方案。文献[8]为提高调度系统对生产环境经常发生变化的自适应能力和全局优化能力,提出了一种基于蚂蚁智能与强化学习相结合的协商策略的多Agent动态调度方法。文献[9]为实现在保持高生产率的前提下对环境变化的快速响应 ,提出了一个基于自适应Agent的模型 ,研究了制造系统中最常见的任务在加工资源上的分配问题。同时还设计了在多智能体系统中分别代表加工资源、任务和协调者的三类Agent ,以及它们在各自利益驱动下所应遵循的行为规则。文献[10]探究各种结构的多Agent系统并分析了它们的优劣。同时还研究了当前应用在多Agent系统的三种主流算法，，dispatching、Scheduling和Pull算法，根据平衡性、扩展性和健壮性三个指标分析了三种算法的适用场景，为多Agent系统的构建和应用提供了很好的指导。文献[11]从制造系统的需求和发展趋势出发，研究基于多Agent的制造车间动态调度问题，通过建立一个基于多Agent的制造车间动态调度原型系统，对Agent理论与技术在智能制造中的应用进行了探索性研究。作者还提出了基于“规则协商”的调度机制和调度策略。将基于规则的调度方法与Agent之间的协商机制相集成，充分发挥各Agent的智能性、自治性和协调能力。文献[12]提出一种短期规划与动态响应能力相结合的混合协商策略。该策略针对柔性车间生产调度问题和面对扰动的再调度问题,将组合拍卖与改进合同网相结合,通过规定相关智能体的动作和交互顺序,实现系统在平稳或扰动状态下的生产调度。基于上述协商策略,开发了一个基于智能体的生产调度原型系统,并通过仿真实验验证协商策略的可行性和有效性。文献[13]为解决制造系统的敏捷生产调度问题，先通过基于功能分解的方法,给出了管理、资源和工件等三类Agent基本组件组成的分布式多Agent调度系统结构、Agent组件基本结构及定义.然后利用基于分布式规则的方法,建立了Agent间的协调策略和调度机制,实现了敏捷生产调度。

面对现代制造业的复杂要求，如果依然遵循传统的调度研究方法，仅仅通过对算法进行改进和扩展来解决是不够的，我们还需要从制造系统整体的角度出发，采用新的思路研究理想的系统模型和合理的系统结构，之后再集成先进有效的调度策略和方法，这样才能适应持续波动的市场需求变化的能力以及响应制造执行过程中出现的生产扰动因素，从而提高企业的竞争力。

1. **基于蚁群算法的动态调度策略**
   1. **蚁群算法的分析**

**2.1.1 蚁群算法的原理**

蚁群算法是一种利用了大自然规律的启发式算法，该算法利用了蚂蚁寻找食物的原理。蚂蚁在运动过程中，会留下一种称为信息素的物质，并且会随着移动的距离，播散的信息素越来越少，所以在食物的周围，信息素的浓度是最强的，而蚂蚁自身会根据信息素去选择方向，信息素浓度越高的路径，蚂蚁选择的概率就越大。而后来选择该路径的蚂蚁又会洒下信息素，进一步吸引更多的蚂蚁，从而找到一条从巢穴到食物的最短路径。同时需要注意的是，信息素本身具有一定的挥发作用。

蚂蚁的运动过程可以简单归纳如下：

（1）当周围没有信息素指引时，蚂蚁的运动具有一定的惯性，并有一定的概率选择其他方向。

（2）当周围有信息素的指引时，按照信息素的浓度强度概率性的选择运动方向。

（3）随着时间推移，信息素会自行挥发。

蚁群算法的原理可以由下图2.1蚂蚁觅食的过程体现出来：



图2.1

图中A代表蚂蚁巢穴，E代表食物。A和E之间存在障碍物，因此当蚂蚁到达B点后，会有两条路径能够到达食物，分别是路径BDE和路径BCE，其中BCE路径较短。初始，当蚂蚁从巢穴出发时，各个路径均不存在信息素，因此蚂蚁选择两条路径的概率都是一样的。但是选择BCE路径的蚂蚁由于路径较短，因此先达到E点，当蚂蚁返回时，由于BCE路径上的信息素浓度更高，因此蚂蚁选择BCE路径的概率更大，而蚂蚁经过BCE的同时又会撒下信息素，进一步提高了BCE路径上的信息素浓度，从而吸引更多的蚂蚁选择BCE路径，最终两条路径上信息素浓度的差异会越来越大，最后会导致所有的蚂蚁都选择BCE路径，如图2.1(b)所示。

**2.1.2 蚁群算法的优缺点**

蚁群算法是一种仿生算法，具有很好的问题求解能力。算法的执行过程是一个从无序到有序的过程，初始，每只蚂蚁都是随机无序地寻找解，随着时间的演变，蚂蚁在信息素的影响下，逐渐遵循一定的规则寻找最优解，最终收敛得到最优解。由于算法的执行是一个正反馈的过程，因此即使随着问题规模的增长，算法也能较快地收敛得到结果。蚁群算法的优点如下：

1. 蚁群算法是一种并行的算法。在搜索过程中，每只蚂蚁的行为都是相互独立的，蚂蚁的行为受环境信息素影响的同时，自身也会撒播信息素。蚂蚁与蚂蚁之间仅仅通过信息素关联起来。因此蚂蚁的集合也可以是一个分布式的多Agent系统，各个蚂蚁独立地进行解空间的搜索，提高了算法的全局搜索能力，也增加了算法的可靠性。
2. 蚁群算法是一种正反馈的算法。显然，蚂蚁的行为是依赖于各个路径上信息素浓度的差异，而蚂蚁最终能够聚集在最短路径找到最优解是依赖最短路径上信息素的累积。最短路径上信息素的累积导致更多蚂蚁选择该路径，而蚂蚁同时也会沿途撒播信息素，又进一步促进信息素的累积，这就是一个正反馈的体现。这个这反馈的过程正是促进了蚂蚁的搜索从无序向有序转变。
3. 蚁群算法是一种稳定性很高的算法。蚁群算法在解空间的搜索过程中基本不需要人工干预，对初始路径的选定要求也不高，算法的执行效率受问题规模增长的影响较小。同时蚁群算法的执行逻辑非常简单，适用于很多组合优化问题的路径搜索和路径规划中。

蚁群算法的应用非常广泛，这证明了该算法的优越性。然而蚁群算法也存在以下一些缺点：

1. 蚁群算法的执行时间较长。虽然蚁群算法的执行逻辑非常简单，执行效率受问题规模增长的影响比较小，但是算法的执行时间相对是比较慢的。这是由于蚁群算法的搜索过程是通过信息素逐渐积累到一定程度，才会有某条路径上的信息素浓度明显高于其他路径从而收敛到最佳路径，这个过程通常是需要比较长的时间的。
2. 蚁群算法可能会过早过快收敛。蚁群算法正反馈的特性能够加快算法收敛得到最优解的速度，但是这个特性可能也会导致过早收敛到某个局部最优解。这是由于可能算法执行的早期阶段，由于蚂蚁的无序搜索可能会有部分蚂蚁聚集到某条局部最优路径上，导致该路径的信息素浓度明显高于其他路径，由于正反馈的特性，最终会吸引所有蚂蚁都得到同样的路径，这样后续的搜索就完全停滞在某个路径上，失去了意义。因此使用蚁群算法求解时，一般都需要采用一定的手段来防止算法过早收敛。
   1. **车间调度中的蚁群算法**

蚁群算法非常适合用于在图中搜索目标路径。根据车间调度的工艺规划特点能够很轻易地将问题转化为对应的图论路径搜索问题。

* + 1. **车间调度中的工艺规划**

传统的车间调度问题大都基于长期的、资源无限的理性前提进行研究，而忽略了实际生产中动态的生产环境、条件约束以及调度过程产生的时间差等等。例如车间调度在实际生产过程中具有突发性和不可预见性，经常会受到设备，人力、原料等资源约束，因此很难以传统车间调度的方式来进行动态调度。

工艺规划是车间调度中的关键核心。生产调度中的工艺规划可以理解为在实际生产过程中，根据生产任务的需要，以时间最短、成本最低等作为目标，确定各设备上所有工件的各道工序在设备上的加工时间以及加工顺序。随着“小批量，多品种”生产模式的普及，实际生产中的工件通常具有多个可选的加工工艺，再加上受工艺约束的工艺排序和机器选择等柔性因素的存在，使得各工件加工具有多条可选的工艺路线。车间调度中的工艺规划通常具有以下特点：

1. 存在工艺约束。某一工件的各道工序有顺序约束关系，只有当当前的工序加工完毕后，后续的工序才能进行。
2. 存在机器约束。在任一时刻，一台设备只能加工一道工序。
3. 工序约束。对于一个工件，任一时刻最多只能有一个工序进行加工。
4. 工件的加工顺序独立。各个工件之间不存在先后顺序之分。这代表不同工件的工序之间不存在加工顺序的约束关系。
5. 通常每道工序都可在两个或两个以上的设备上完成，分别对应不同的完成时间。

工艺流程图代表一个完整的工件的完成需要经历的加工工序。如图2.2所示，是两个工件的工艺流程图。图中每个结点代表一道工序或者工序间的约束关系。



图2.2

如图2.2(a)所示，1号结点代表一道工序，表示该工序可在2号或3号设备上进行加工，所需要的时间分别是18分钟和9分钟。3号结点是一个与（AND）节点，表示工序2加工完成后，从与节点引申出来的两条加工路径都需要完成。与3号结点对应的是汇合(JOIN)结点，JOIN结点表示只有当工序5和工序6均完成后才能进行下一步的工序加工。在图2.2(b)中，2号结点是或（OR）结点，表示从2号结点引申的两条分支路径仅能从中选择一条。在实际生产中，正是由于工艺规划中存在多条工艺路径，而且随着问题规模的增长，工艺路径的数目会呈指数式增长，因此动态调度过程是一个典型的NP难题，而蚁群算法的优良特性能够很好解决这个问题。

* 1. **车间调度中蚁群算法的数学模型**

当我们把多个工件的工艺流程图的起点（S结点）连接起来，终点（E结点）也连接起来，就能组成一个有向图，如图2.3所示。这时，我们寻找完成时间最短的调度方案就能转换为在有向图中寻找一条路径最短的哈密顿回路，其中我们把每道工序的完成时间形象化为有向图中工序间的边，即距离。由于蚁群算法非常适用于路径搜索，因此我们只需要根据车间调度的特点对基本蚁群算法进行少量的修改就能很好地适用于动态调度的过程。



图2.3

* + 1. **车间调度中蚁群算法的参数描述**

1. 工序规模为n，即多个工件共包含有n道工序，从0 ~ n - 1对其进行编号。若i，j为其中的任意两道工序的编号，则有i,j>=0&& i , j <= n - 1。
2. 蚁群规模为m，即蚁群中有m只蚂蚁，从0 ~ m - 1对其进行编号；k为其中的任意1只蚂蚁的编号，则有 k >= 0 && k <= m - 1。
3. dij：两道工序i和j之间的距离。将任意两道工序看作两个点，则此距离为二维平面上这两点间的直线距离。设定多道工序不可重叠在同一个点上，因此当i = j时，dij=0，但此时无意义；当i != j时，dij>0。又规定两道工序间的距离向量是对称的，即有dij=dji。
4. ηij：蚂蚁从工序i转移到工序j的启发程度，即环境因素本身对蚂蚁选择判断的影响，这个值在蚁群系统的运行中不改变。在本算法中ηij=1/dij，即i，j两城市的距离越短对蚂蚁选择的吸引力越大。又因dij=dji，则同样也有ηij=ηji。
5. 时刻t，为计算机模拟的实际环境时间。该时刻是离散而非连续的，从0时刻起，最小离散单位为1，即时刻为0，1，2，3，4，5···通常在1个时刻以内会发生蚂蚁移动一步这类的原子事件。
6. bi(t)：t时刻位于工序i的蚂蚁数目。有。
7. τij(t)：工序i，j之间t时刻时边上的信息素轨迹强度，初始t=0时刻时，设置一个初始常量τij(t)=C，其中C∈N。τij(t)∈N；当i=j时无实际意义；又因边具有对称的特性，故τij(t)=τji(t)。
8. Δτkij(t,t+Δt)：在t时刻到t+Δt时刻这Δt的时间内，蚂蚁k在（i，j）工序间路段上产生的信息素量。同样，Δτkij(t,t+Δt)∈N且Δτkij(t,t+Δt)=Δτkji(t,t+Δt)。
9. Δτij(t,t+Δt)：在t时刻到t+Δt时刻这Δt的时间内，（i，j）工序间路段的信息素增量。有。同样，Δτij(t,t+Δt)∈N且Δτij(t,t+Δt)=Δτji(t,t+Δt)。
10. τij(t+Δt)=ρ⋅τij(t)+Δτij(t,t+Δt)，其中ρ∈[0,1]为τij(t)在刷新信息素时的保留百分比。若为0则全部挥发，若为1则全部保留。
11. tabuk：蚂蚁k的禁忌表。初始时为空。每当蚂蚁到达一个工序都将该工序放到禁忌表中。由于工艺规划的特点是每个工序只会被加工一次，当蚂蚁走到有向图的终点时，禁忌表中的工序顺序组成的路径即为动态调度的一个解。
12. accessiblek，蚂蚁k的可行池。初始时为有向图起点的所有后续结点。每只蚂蚁的可行池代表蚂蚁下一步可以达到的工序。由于工艺规划的工序约束，同一工件中的工序时存在先后顺序约束的，同时由于与结点、或结点的存在使得这种约束关系更加复杂，因此使用可行池来记录蚂蚁下一步可到达的工序。每当蚂蚁达到一个工序，把工序放入到禁忌表的同时，也要把该工序的后续工序放入到可行池中。
13. Lk：第k只蚂蚁在求得某解后所走过的路径长度，对应的就是调度方案所需要的时间。
14. Q：蚂蚁在求得一个解后在其路径上播撒的信息素总量。这是一个常量值且Q∈N+
15. 蚂蚁的运动速度与路径的实际长度无关；固定每个1个单位时间内可从某工序到达可行池中国任意另一道工序。
17. ，j∈accessible
18. bestWay：每轮找到的最优路径组成的数组；从当轮蚂蚁填满的禁忌表中筛选。标准为走过的长度Lk最短。
19. Ncmax：常量，最大求解循环次数，当到达该次数后算法停止。Ncmax∈N+。

**2.4 车间调度中的蚁群算法流程**

循环层1：共循环Ncmax轮，每轮循环用时(n-1)个单位时间并发生如下事件：

本轮刚开始的时刻为t时刻，生成一个规模为m的蚁群，将m只蚂蚁随机放置初始工序S中，并将蚂蚁们的初始工序放入其禁忌表中。该操作不耗费时间。

（1）循环层1-1：循环n-1次，该循环结束后每只蚂蚁的禁忌表都将被填满，每只蚂蚁都成功地达到终点E，时间流逝了n-1个单位时间,每次：

1.时间t=t+1,表示每次循环时间都流逝了1个单位时间。

循环层1-1-1：循环m次，这m次循环是在一个单位时间内同时并行发生的，每只蚂蚁都做如下动作：

蚂蚁依Pkij(t)到达新的工序，同时将新的城市加入自己的禁忌表，并把后续工序放到自己的可行池。

（2）循环层1-2：循环m次，遍历所有蚂蚁的禁忌表；得到每只蚂蚁的Lk；得到本轮的最优路径bestWay；将蚂蚁禁忌表中走过的路径播洒上信息素。Δt=n−1

（3）将本轮的最优路径bestWay与全局的最优路径对比，若更优秀，则替换。

（4）消灭当前的蚁群（杀死所有蚂蚁），然后休整一个单位时间，t=t+1。

**4 参考文献**

[1]李京生. 面向多品种变批量生产的重构调度方法[D].北京理工大学,2014.

[2]Rangsaritratsamee R, Ferrell W G, Kurz M B. Dynamic rescheduling that simultaneously considers efficiency and stability[J]. Computers & Industrial Engineering, 2004, 46(1):1-15.

[3]Zhang L. Distributed Genetic Algorithm for Integrated Process Planning and Scheduling Based on Multi Agent System[C]// Manufacturing Modelling, Management, and Control. 2013:760-765.

[4]Nguyen V, Bao H P. An Efficient Solution to the Mixed Shop Scheduling Problem Using a Modified Genetic Algorithm [J]. Procedia Computer Science, 2016, 95:475-482.

[5]王进峰,阴国富,雷前召,张超. 蚁群算法在工艺规划与车间调度集成优化中的应用[J]. 东南大学学报(自然科学版),2012,(S1):173-177.

[6]Qi J G, Burns G R, Harrison D K. The Application of Parallel Multipopulation Genetic Algorithms to Dynamic Job-Shop Scheduling[J]. The International Journal of Advanced Manufacturing Technology, 2000, 16(8):609-615.

[7]廖强,周凯,张伯鹏. 基于现场总线的多Agent作业车间动态调度问题的研究[J]. 中国机械工程,2000,(07):45-47+5.

[8]徐新黎,郝平,王万良. 多Agent动态调度方法在染色车间调度中的应用[J]. 计算机集成制造系统,2010,(03):611-620.

[9]王晓芳,杨家本. 制造系统中任务分配的自适应Agent模型[J]. 计算机集成制造系统-CIMS,2001,(08):17-20+58.

[10] Baker A D. A survey of factory control algorithms that can be implemented in a multi-agent heterarchy: Dispatching, scheduling, and pull[J]. Journal of Manufacturing Systems, 1998, 17(4):297-320.

[11]宋娟. 基于多Agent的制造车间动态调度系统研究[D].沈阳工业大学,2004.

[12]吕赐兴,朱云龙,尹朝万. 基于Agent的生产调度混合协商策略研究[J]. 计算机集成制造系统,2006,(12):2074-2081.

[13]王艳红,尹朝万. 一类基于多Agent和分布式规则的敏捷生产调度[J]. 控制理论与应用,2004,(04):526-530+536.