

摘 要

刀具流调度是柔性制造系统(FMS)调度的重要组成部分, FMS 的生产效率和系统柔性在很大程度上取决于刀具流的合理规划。当前刀具流的调度理论与算法还十分有限, 实际运行中的刀具流自动化水平也较低, 这种现状已经在较大程度上影响了 FMS 生产能力的进一步发挥, 因此, 为了提高系统的总体加工性能和充分发挥 FMS 本身的柔性, 进行更深入的刀具流调度研究具有重要的理论价值和实际意义。

本文以 FMS 中刀具流调度为研究对象, 分析了刀具流动态调度研究的必要性, 回顾了刀具流调度理论的研究现状, 研究了 FMS 生产中工序备刀与工步备刀两种情况下的刀具流调度问题, 并对刀具流调度过程中的死锁检测和死锁避免等关键技术问题进行了深入分析。

首先, 结合面向对象技术和 Petri 网理论分别为 FMS 生产中的工序备刀情况和工步备刀情况建立了调度仿真模型。该模型不仅包括了工件流, 同时还重点考虑了刀具流进程, 并将机床故障与刀具失效作为一个组成部分包括进来, 从而准确方便地描述了考虑刀具流的 FMS 运行各个环节, 为后续的刀具流启发式调度研究奠定了基础。

其次, 在以往工序备刀前提下的刀具流无死锁调度工作基础上, 进一步将刀具寿命考虑进来, 从而丰富和完善了现有刀具流相关调度理论与技术。采用图论方法建立了扩展的刀具申请分配图, 并将其和刀具流死锁联系起来, 从图论角度深刻揭示了刀具流死锁本质。在扩展的刀具申请分配图基础上, 建立了刀具流死锁判定定理和非死锁图的性质定理。提出了两层死锁避免策略, 分别用于机床选工序和刀具分派两模块的死锁控制, 实例分析表明了该策略的合理性和可行性。

在刀具分派决策中充分考虑了刀具流对工件流效率的影响, 提出了目标机床当前加工完成时间最早规则。实验结果表明该规则能够有效地减少机床的刀具等待时间, 提高了系统的生产效率。将两层死锁避免策略与调度规则相结合, 最终建立了工序备刀下的刀具流无死锁启发式调度算法, 并给出了刀具失效的处理策略, 仿真结果表明了该算法的有效性。

在完成了工序备刀下的刀具流研究之后, 基于图论方法深入研究了工步备刀前提下刀具流的无死锁调度问题。建立了刀具流状态演变模型, 给出了刀具流死锁状态及无死锁不安全状态的定义。构建了刀具资源分派图并提出了机床死锁的概念, 将机床死锁与刀具流的无死锁不安全状态有机联系起来, 设计了一种刀具流无死锁不安全状态检测算法, 并在此基础上进一步给出了刀具流的死锁避免方法。实例表明该方法能够简单有效地避免死锁的产生。

在刀具分派决策研究中提出了虚拟机床队列思想, 并将其与死锁避免方法相结合, 最终构建了工步备刀下的刀具流无死锁启发式调度算法, 并对调度过程中的刀具

失效给出了相应的处理策略，解决了工步备刀下的刀具流死锁避免和刀具实时分派问题。

在完成了工步备刀下的刀具流无死锁启发式调度研究之后，进一步从全局优化角度分析了刀具流调度问题。以系统完工时间最少为优化目标，建立了工件流和刀具流综合调度数学模型。提出了一种双重遗传算法对优化模型进行求解：外层遗传算法用于求解可行工序加工序列；内层遗传算法用于最优可行刀具分派方案的搜索。实例分析表明了双重遗传算法的有效性和可行性。为使调度系统能够有效处理机床故障和刀具破损等突发事件的再调度问题，进一步给出了一类基于双重遗传算法的准静态优化再调度方法。

最后，建立了各类算法的仿真调度程序，对各算法下系统的性能指标进行了比较和综合评价，仿真分析结果表明所建立的工序备刀下的启发式调度算法和工步备刀下的启发式调度算法及优化调度算法是可行而有效的。这些调度理论与算法的提出为 FMS 的设计和运行提供了依据，为实际生产中 FMS 性能的充分发挥奠定了基础。

关键词：柔性制造系统；刀具流；死锁检测；死锁避免；双重遗传算法

Abstract

Production efficiency and system flexibility of Flexible Manufacturing System (FMS) are mostly depend on reasonable planning for tool flow, whose schedule is one of important elements of FMS scheduling. But the state-of-the-art scheduling theory and algorithm of tools flow are not so advanced that the degree of automation of tools flow in job shop is comparatively low, which greatly restrain the production ability of FMS increasing. Hence, further research on proper planning and scheduling for tool flow is imperative from the view of theory and practice in order to improve the overall system production ability and make full of the advantage of FMS in flexibility.

In the course of the thesis presented, the subject of research is tool flow scheduling in FMS environment. The necessity of tool flow dynamic scheduling research is stated, and the state of the art of tool flow scheduling theory is reviewed. Then tool flow scheduling in cases of procedure provisioning tools and working step provisioning tools are investigated, and the deadlock detecting and deadlock avoidance during the schedule of tool flow are deeply analyzed.

Firstly, simulation models are developed respectively in cases of procedure provisioning tools and working step provisioning tools based on Petri net with the oriented-object technique. The models developed accurately and conveniently describe all the sections when FMS is running with not only considering the part flow but also focusing on the tool flow, and include the machine malfunction and tool failure as a component. This is basic for tool flow heuristic scheduling.

Secondly, cutting tools life is considered as a factor to rich and enhance tool flow scheduling theory and technique based on tool flow deadlock-free scheduling on the condition of procedure provisioning tools. Extended tools applying allocation graph in relation to tool flow deadlock strategy is built up with the graph theory. The nature of tool flow deadlock is revealed profoundly from the view of graph theory. Tool flow deadlock decision theorem and non-deadlock graph property theorem are given based on extended tools applying allocation graph. Double-layer deadlock avoidance strategy is brought forth to prevent deadlock in two models of machine selection process and tools allocation.

The effect of tool flow on part flow efficiency is fully considered in tools allocation decision, and the rule of earliest finish time for current operation is given. The experiment results show that the rule presented is able to decrease the waiting time for tools effectively and improve the production efficiency of the system. Tool flow deadlock-free heuristic scheduling algorithm in case of procedure provisioning tools is built up with the method of

combination double-layer deadlock avoidance strategy and scheduling rules, and the handling tactics of tools failure are given. The simulation results show that the algorithm is effective.

Then tool flow deadlock-free scheduling in case of working step provisioning tools based on graph theory is researched deeply. The evolving model of tools flow state is built up, and the states of tools flow deadlock and tools flow deadlock-free unsafe are defined. The tools resource allocation graph and the definition of machine deadlock are given. Combined machine deadlock with tools flow deadlock-free unsafe state, a detection algorithm of tools flow deadlock-free unsafe state is designed, based on which the tools flow deadlock avoidance approach is developed. An example shows this approach is able to avoid the occurrence of tools flow deadlock.

Tool flow deadlock-free heuristic scheduling algorithm in case of working step provisioning tools is developed based on the virtual machine queue idea and the deadlock avoidance approach in tools allocation decision, then the corresponding handling strategy of tools failure in the progress of scheduling is given. Thus, tool flow deadlock avoidance and tools allocation in real time are solved in case of working step provisioning tools.

Tool flow scheduling is analyzed from the view of the global optimization, after research on tool flow deadlock-free heuristic scheduling. A mathematical model of integrated scheduling for part flow and tool flow is presented, in which the objective is to minimize system make-span. A double-layer genetic algorithm (GA) is proposed for global optimization of the model. The outer and inner layers of the GA are to search optimal and feasible operation sequences and tool assignment, respectively. A case study shows that the double-layer GA is effective and practicable. The Quasi-static optimal rescheduling base on double-layer GA is presented in order to deal with effectively emergencies such as machine malfunction and tools breakage.

Finally, the scheduling simulation programs based on algorithms developed are given and the system performance index by means of the algorithms. The simulation results show that the heuristic scheduling and optimal scheduling algorithms in case of procedure provisioning tools and working step provisioning tools are feasible and effectively. The presented scheduling theory and algorithm provide a foundation for FMS design and running, and establish a basis for fully utilizing FMS performance in actual production.

Keywords: flexible manufacturing system (FMS); tool flow; deadlock detection; deadlock avoidance; double-layer genetic algorithm

第1章 绪论

1.1 研究背景

制造企业的竞争力与产品的供货周期、质量和售后服务水平密切相关。先进的生产管理是实现竞争力的一个重要手段。生产调度作为生产管理的核心内容和关键技术，其目标是在企业车间有限的资源约束下，合理制订工件在相关设备上的加工顺序和加工时间，以达到所选定的生产性能指标最优。由此可知，科学地制定生产调度方案，对于提高产品交货期满足率，缩短产品供货周期、控制车间的在制品库存和提高企业生产率起着至关重要的作用。

在传统上，生产调度的研究需要以一些公认的假设为前提，这些假设一般包括：1) 确定的被调度的工件集合；2) 确定的工件的工时，并且在安排计划时全部工件都已就位；3) 用于加工工件的设备资源是可以连续工作的。这类调度问题归于静态调度问题^[1]。

与上述理想状况不同，实际生产中的很多情况往往是随机发生的，例如工件会随机到达，机床会随机发生故障，刀具会随机发生破损等，这些随机事件产生的扰动将使得预先规划好的静态调度方案难以正常执行。因此，合理的生产调度应该能够在加工过程中根据状态的实时变化，对生产活动进行动态地、实时地和持续地进行优化控制，从而保证系统在加工过程遇到故障和扰动时，能够根据系统监控到的实时状态自主修改原定的加工顺序，优化调度系统的所有资源，使FMS持续地、稳定地运行。在研究中，这类问题通常被称为动态调度问题。

早在上世纪50年代就出现了动态调度的概念。1957年，Jackson^[2]便明确区分了静态调度和动态调度的概念。最初对此研究的学者主要采用了启发式方法或仿真方法。近年来，随着计算机技术的迅速发展，推动了人工智能、遗传算法、神经网络和仿真技术等新方法的产生和发展，为动态调度的研究提供了新工具、开辟了新思路。与此同时，也为生产调度的实用化奠定了基础。由于生产发展的不断需求，动态调度当前已成为生产调度研究中的热点之一。

按调度对象的不同，生产中的动态调度分为工件流动态调度和刀具流动态调度两类。

工件流动态调度的对象主要包括系统内待加工或加工后的工件以及机床、工件运输小车（AGV）、托盘和缓冲区等设备资源。动态调度的内容主要包括工件的加工顺序及设备资源的服务顺序等^[3-4]。这一类问题的研究较为深入而广泛，采用的研究方法主要有Petri网方法^[5-9]，遗传进化算法^[10-11]，神经网络法^[12-13]，基于知识的调度方法^[14-16]

以及其他一些启发式调度方法^[17-19]和自适应算法^[20]等等。

刀具流动态调度主要包括刀具需求计划和刀具实时分派两方面内容。刀具需求计划是指对于给定的一批加工工件，系统应该配置多少类刀具，每类刀具应配置多少把，才能满足工件加工的需要或者达到较高的生产率等问题。刀具实时分派是指在刀具流不发生死锁的前提下，如何在决策点为各加工机床分派刀具，才能得到较好的系统性能，其中包括了刀具决策、刀具运输、刀具失效处理及刀具流死锁检测与避免等诸多子问题。

在FMS运行中，刀具流和工件流同时存在并相互依附。工件流驱动刀具流为之准备所需刀具，而有限的刀具资源又制约着工件流的各项决策和运行过程，所以对刀具流动态调度的研究包括刀具实时分派的同时又不可避免地必须将工件的动态排序包括进来。由于长期以来对工件流研究比较集中而忽略了对刀具流的研究，所以目前刀具流的调度理论还比较薄弱，实际运行中的刀具流自动化水平也较低，相关文献主要集中在对刀具流静态调度^[21-30]的研究及不考虑工件流的刀具流动态调度研究^[31-36]，这种现状已经在较大程度上影响了FMS生产能力的进一步发挥，因此，为了提高系统的总体加工性能和充分发挥FMS本身的柔性，进一步深入研究刀具流的调度是十分必要的。

1.2 刀具流动态调度研究现状综述

1.2.1 刀具流的作用及其物理模型

(1) 刀具流的地位与作用

刀具作为生产制造系统的重要加工资源，对系统的生产率和利用率起着重要的作用，同时刀具的购买成本比较大，约占系统总费用的20%~30%^[37]。另外由于生产加工的需要，刀具频繁的在各机床之间及机床与中央刀库之间进行交换和流动，因此迫切需要一个功能完善的刀具管理系统对刀具进行管理和调度，以实现系统中刀具资源的充分利用。

FMS中的机床设备一般都配备有刀具库，以存储当前加工所需的刀具并实现加工过程中的快速换刀。每台机床刀具库的容量一般为30~60把，有些大容量刀具库可容纳刀具70~100把^[37]。FMS在早期建立的加工范围不大，机床刀具库存储的刀具基本上都能满足加工的需求，因此刀具的交换和流动不大，系统对刀具的调度和信息管理的要求不高。随着社会对产品多样化、低加工成本及短制造周期等需求日趋迫切，要求现代的FMS具有更高的柔性，能够实现多品种工件的加工。同时，为降低FMS的生产投入成本，机床刀具库的容量也减少了，所以仅仅依靠有限的机床刀具库所存储的刀具是不能够满足一台机床加工的所有用刀需求的^[38]，故解决的办法是增加中央刀库

或者实现各机床刀库之间刀具的高度共享,在这种情况下, FMS中刀具流调度和控制问题逐渐凸显出来。

与已有的工件流调度控制理论相比较,现有的刀具流控制理论还很少很不成熟,所导致的刀具管理技术水平的低下直接影响了FMS的生产效率、生产质量和柔性的进一步发挥^[39-40]。Gray, Seidmann和 Stecke^[41]等人指出,刀具流调度理论的薄弱导致了实际生产中FMS运行困难,阻碍了FMS生产能力的进一步发挥。Carrie和Perera^[42]认为,有效的刀具管理可以提高系统的经济效益。Suri和Whitney^[43]则强调指出了刀具流调度在FMS的调度控制中占有十分重要的地位。总之,人们对刀具管理系统以及刀具调度理论的研究在不断的深入。由此可见,刀具流作为柔性制造系统的一个重要组成部分,在FMS的生产过程中占有十分重要的地位,它的管理是否完善,调度是否合理,控制是否有效,对整个柔性制造系统的利用率、生产效率、生产质量和加工成本都将产生很大的影响^[44-45]。

(2) 刀具流的物理模型

通过分析国内外现有的FMS刀具流系统^[46-50],可以发现它们的组成一般都包括:

1) 刀具管理计算机:计算机刀具调度系统软件是刀具调度系统的大脑部分,负责指挥、操作整个制造系统的刀具调度、运送、刀具购买计划等;

2) 刀具进出站:人与刀具自动输送系统的连接点。设有多个存放刀具的刀位,每个刀位都设有监控开关。新刀由此进入系统,磨损、破损的刀具由此输出;

3) 刀具输送网络:主要采用自动引导小车或刀具机械手等装置完成正常或异常(如磨损、破损)刀具的输送和更换;

4) 刀具存贮区:包括局部机床刀库和中央刀库。机床刀库刀具容量有限,一般存放30-60把刀具,主要存储本机床加工零件所需的全部或部分刀具,刀库结构多为链式,容量较大而且容易扩充。中央刀具库的容量很大,可容纳数千把各种刀具,主要用于储存机床加工零件的各种刀具,机床不急需使用的刀具,加工某工序所需的特殊刀具及各种刀具的备用刀具等。

图1.1所示的刀具流物理模型概括了各类刀具流系统的总体结构。在此模型中,FMS由M个柔性制造单元(Flexible manufacturing cell, FMC)构成,每个FMC由N台机床组成。该模型的刀具流包括以下内容:

预调好的刀具由刀具进出站进入系统,然后由刀具输送装置按刀具分派表将刀具送至每个机床的局部机床刀库。局部机床刀库用来装载目前加工所需的刀具,它的配置是根据目前机床的加工零件、加工负荷实现的。机床按被加工工件的要求,可以直接通过刀具输送装置从刀具进出站、中央刀具库或其它机床的局部机床刀库调取所需要的刀具。在实际加工过程中,机床可一边加工,一边往自己的刀具库调用要求的刀具。中央刀具库是系统级刀具库,主要用于贮存系统中加工工件所需的各种刀具及备用

刀具。任何破损、磨损的刀具从机床刀库返回到中央刀库，进行修缮之后继续投入系统使用。将中央刀具库通过刀具自动输送装置与刀具进出站及各局部机床刀具库连接起来，就可形成自动刀具供给系统。

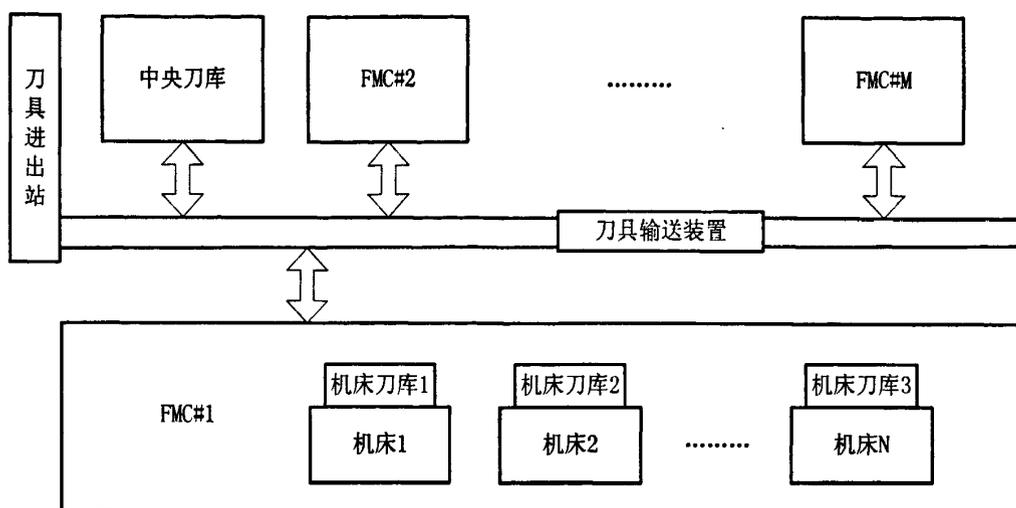


图 1.1 刀具流物理模型

Fig.1.1 Physical model of tool flow

1.2.2 刀具流调度建模方法研究综述

与工件流研究相比较，刀具流的研究时间还很短，相关文献主要集中在介绍刀具管理系统的功能、体系结构及状态监控^[51-57]，而与实际加工过程相关的刀具调度理论研究^[38,58-59]则不多见，这主要是由于FMS趋向于小批量复杂零件的生产，刀具的需求数量和种类非常大，刀具流的控制和决策相当复杂，故对该领域的研究具有一定的难度。因此，对刀具流进行更深入的研究具有重要的理论意义和工程实用价值。

刀具流的研究首先应从建模开始，只有针对纷繁复杂的刀具系统做高度的抽象，提取其本质要素，忽略其次要因素，构建出合理的模型，才能够更好地开展进一步调度研究。常用的建模方法主要可分为以下三类：

(1) 基于petri网的建模方法

为了全面形式化的描述柔性制造系统的运行状态，许多研究人员都采用Petri网为FMS建模。C. Reddy^[60]采用时间Petri网对中心刀具管理系统进行建模，并研究了刀具预定时间和刀具数目对系统完工时间的影响，但其研究是在分级控制模式的假设下进行的，且刀具不能在机床间共享，限制了系统的灵活性；K. BARKAOUI^[61]采用随机petri网进行建模，在此基础上对机床的利用率及刀具运输时间进行了仿真评估。但其刀具共享策略是整批刀具更换，增加了刀具资金投入的同时也限制了机床共享刀

具的灵活性。F.F. Chen^[63]针对动态刀具分派下的多级多机柔性制造系统,采用面向对象的着色petri网进行建模,大大降低了系统的复杂性。并对不同刀具需求选择规则下的系统性能进行了评估。毕诸明等^[62]为预测被测FMS运控软件中刀具流引起的死锁和回避功能。首先建立了刀具流的PN模型,然后从网模型中变迁的等待关系来定义FMS刀具流的死锁现象,但是该模型只局限于刀具流内部,没有考虑工件流对刀具流的影响。舒海生等^[64]运用面向对象技术和高级Petri网,针对工序备刀情况为柔性制造系统建立了一个含机床故障处理的动态调度仿真模型,该模型层次分明,故障处理流程十分简洁,较好地反映了系统的运行情况,但模型中未包含刀具失效的处理。

可见,petri网作为一种形式化的建模方法可以很好的描述柔性制造系统中可能发生的各种变化以及变化间的关系。然而,随着制造系统中工件和资源数量、种类的增加,采用Petri网方法进行分析将引起系统控制模型的复杂化,不利于构建高效的实时控制算法。

(2) 基于图论的建模方法

为了描述柔性制造系统中工件和资源的交互关系,一些研究人员采用了图论方法为FMS建模。Housni Djellab^[65]针对工件加工顺序和刀具的载入顺序问题,采用超图法进行描述,并用启发式算法求解,目标是最小化换刀次数。GEBRAEEL^[66]针对刀具共享策略利用图论方法建立了刀具资源分派图,图中顶点集合分别与工件资源和刀具资源相对应,边的集合表示了工件对刀具资源的需求和刀具资源对工件的分派,权集表示了工件对刀具的占有和释放顺序,该图用简洁精确的方式描绘了系统中刀具的分派和需求,但该模型描述只局限于刀具流内部。LEE C S^[67]针对FMS中工件加工顺序和刀具选择问题,对每个工件的多个可行的操作顺序建立了网络结构图,图中每个节点包含着一个操作号及其加工所需的刀具和加工时间等信息,弧代表两个操作之间的先后关系,然后以最小化等刀时间为目标,通过左移和右移操作选择一个可行的操作顺序。

图论方法简单直观,可以清晰地反映系统中工件和资源在当前状态下的交互状态信息,但是该方法不适合于系统的动态行为描述。

(3) 基于整数规划的建模方法

一些研究者基于运筹学原理,采用整数规划的方法为系统建模。T. S. Atan^[68]对机器分派和刀具分配问题,以最小化刀具配置数量和最大化刀具利用率为目标,以刀具存储区容量为约束条件,采用整数规划方法进行建模。Atmani A^[69]针对FMS中的机器-刀具分配和操作分派问题,以最小化总的成本为目标,以刀具库容量和刀具寿命为约束条件,建立了混合整数线性规划模型,且一个操作可以在不同的机床上加工,即考虑了可替代机床情况;Gamila M A^[70]针对工件刀具的载入和调度问题,以机床刀具库容量和刀具寿命为约束条件,建立混合整数线性规划模型进行工件刀具的初始分派,

然后根据模型的输出结果进行工件的调度。S. G. Ponnambalam^[71]和Rahul Swarnkar^[72]针对FMS中的机床选择、工序分派和刀具需求问题,以最小化系统的不平衡度和最大化系统的输出为目标建立了数学模型。S. Bilgin^[73]针对FMS中的操作运行分解的情况下的调度分派问题,以加工操作总的全重为目标,以机床加工能力、刀具库容量和刀具拷贝数量为约束条件,建立混合整数规划模型。

可见,利用整数规划对系统建模可以分析系统的生产率、平均流通时间等性能指标,但它很难将车间的其它资源(如:AGV、刀具机械手等)考虑进去,并且很难处理系统出现异常时的情况。

综上所述,现有刀具流的建模一方面大多局限在刀具流内部,而工件流对刀具流的影响则考虑得较少;另一方面基本上只考虑了系统处于正常状态下刀具流的流动,没有包含对故障(刀具失效等)的处理。事实上,在FMS运行中,工件流和刀具流是同时存在并相互影响的,在刀具流建模过程中应该考虑到与工件流进程之间的相互作用,而刀具失效等异常情况在系统运行中也是时有发生,因而在建模时也应予以考虑。此外,现有刀具流(包括工件流)的研究大多将工序加工不可中断这一前提作为建模的出发点,而事实上在实际加工过程中,尤其是在高度自动化的加工中心上,采用的是工序集中原则,各工序往往包括了很多的工步,加工时间很长,因此,如果仍然将工序不可中断作为建模和调度的前提将会对刀具的利用率造成不利影响,而如果将工序可中断(即以工步为备刀对象)作为出发点,应当可以取得更好的效果,这一方面的研究目前还很匮乏。

1.2.3 刀具流动态调度方法研究综述

在对考虑刀具流的制造系统建立模型后,就要构造相应的调度算法进行问题求解。刀具流动态调度的一般研究方法可以分为两大类:“仿真+规则”启发式调度和最优化调度,其中最优化调度又包括传统优化调度和智能优化调度两种。

(1) “仿真+规则”启发式调度方法

由于FMS加工的工件种类繁多,系统运行时需要的刀具种类和数量是很多的,而且这些刀具频繁地在各机床之间、机床和刀库之间进行自动交换。另外,刀具磨损和各种偶然因素引起的刀具早期失效所造成的强制性或适应性换刀,使得刀具流的规划和调度变得异常复杂,常规的规划和调度方法是难以胜任的,不能充分地发挥FMS的生产效率和高柔性等生产潜力,因此人们往往采用“仿真+规则”的方法来解决这一问题。

Chang-Yong^[74]分析了FMS中工件的载入和刀具分派问题,以减少机床间刀具交换次数为目标建立一种启发式调度算法,同时在调度过程中考虑了可替代刀具情况。

Arun S. Kashyap^[75]讨论了FMS中,在不同的刀具配置下刀具选择规则和刀具分派规则

对系统完工时间和刀具机械手利用率的影响。H.K.Roh^[76]针对机器载入、刀具分派和工件加工顺序问题,以最小化拖延期为目标,提出三个启发式算法。Choong Soo Lee^[77]针对FMS中的动态刀具分派策略,提出了考虑刀具流调度的工件分派模型,该模型包括两部分,即工件选机床模型和机床选工件模型。目标是最大化系统生产率及最小化拖延工件的数量。Housni Djellab^[65]针对工件加工顺序和刀具的载入顺序问题,采用超图法进行描述,目标是 minimized 换刀次数,并用启发式算法求解。Y. Fathi^[78]针对并行加工环境中工件和刀具的调度问题提出三个启发式算法,并以系统完工时间为目标对三种算法进行了仿真分析。

Kwasi A.G^[79]讨论了FMS中刀具共享情况下的刀具分派问题,针对三种不同的刀具分派策略,提出相应的启发式算法,通过仿真考查它们对交货期、拖延工件百分比和工件平均流通时间的影响,但是没有包含订单到达、机床刀具故障等紧急情况的处理。Ozbayrak M^[80]针对FMS中工件流和刀具流的调度问题进行仿真研究,并对4种刀具载入策略和4种工件调度规则组合下系统的性能指标进行了分析,但是没有考虑可替代工艺路线、可替代机床和可替代刀具情况。BUYURGAN N^[81]针对FMS中存在可替代刀具情况下的刀具选择问题提出一种启发式算法,所提的算法以刀具尺寸和刀具寿命的比率L/S为依据进行刀具选择和分派,实例证明了算法的有效性,但是对刀具失效等动态情况没有考虑。

SURESH K N^[82]研究了单级多机柔性制造系统中的刀具共享问题,对三种不同刀具载入策略下的系统性能进行了比较分析,但仿真中没有考虑机床故障和刀具破损等故障情况的处理;ZHANG Xiaoming^[83]和SURESH K N^[84]分别对工件移动策略和刀具移动策略对系统性能的影响进行了比较研究,而对两种策略同时存在时系统性能情况没有分析。SURESH K N^[85]针对FMS中无故障和有故障(机床故障和AGV故障)情况系统的性能进行了仿真,但是刀具移动策略只针对一个柔性制造单元内,限制了系统内机床共享刀具资源的灵活性。舒海生^[64]针对工序备刀下的刀具流问题,建立了基于刀具预准备法的启发式调度算法,仿真结果表明该启发式调度算法取得了较好的系统性能,但在调度过程中未考虑刀具失效对系统的影响。

可见,“仿真+规则”的调度方法针对车间调度问题的NP特性,并不试图寻求问题的最优解,而是在调度效果和计算时间之间进行了折中处理,力图以较小的计算量来得到满意解或近优解。由于该方法具有计算量小、效率高、实时性好的特性,因而在动态调度研究中被广泛采用^[1]。

综上所述,已有的刀具流启发式调度研究集中于刀具流内部,较少考虑工件流对刀具流的制约作用,同时在调度过程中未考虑刀具失效的影响。事实上,工件流和刀具流是柔性制造系统的两个不可分割的有机组成部分,二者是相互影响的,共同构成一个有机统一体,因此在动态调度中只有将工件的加工排序和刀具的分派同时进

行,才有可能取得全局的较优的调度。此外,刀具失效是调度过程中不可避免发生的,如果忽视了对它的处理将会使得原有调度过程陷入混乱,因此在调度算法中也应加以考虑。另外,现有的一些调度策略和规则只局限于工件流或刀具流内部,没有考虑二者的相互作用和影响,不利于改善和提高系统综合性能,故还需要建立着眼于系统整体性能的新的调度规则。

(2) 最优化调度方法

1) 传统优化调度

传统优化调度方法首先建立调度问题的数学规划模型(动态规划模型、非线性规划模型以及整数规划模型等),然后根据所建模型采用相应的求解方法,如:逐次逼近法、分支定界法及割平面法等。Gamila M A^[70]在考虑机床刀具库容量和刀具寿命时,采用分解的方法解决机器载入和调度问题,首先建立一个MILP模型进行工件操作的初步分派,同时确定刀具的数目及其分派的机床,然后再提出一个启发式算法进行工件的调度。但其是在调度以前进行机器的载入和刀具的分派,没有考虑任务的实时决策问题,且每类刀具只允许一个刀具拷贝。MELTEM^[87]针对FMS中等规模短期的生产计划中工件分派问题,以机床有效工作时间和刀具库容量为约束条件,建立混合整数规划模型,提出分支定界法对模型求解。A. Grieco^[88-89]针对FMS中工件类型选择和静态刀具分派问题,以最小化刀具拷贝的数量为目标,建立了整数规划模型,并用割平面法对模型进行优化求解。M. SELIM^[90]针对刀具分派和加工条件优化选择问题,以最小化生产成本为目标,以刀具可获得性,刀具寿命,刀具分派,加工条件为约束建立了非线性的混合整数规划模型,并提出一个分解算法对模型进行求解。Selin^[91-92]针对FMS中的操作和刀具分派问题,以最大化总的权重为目标,以机床加工能力、刀具库容量和刀具拷贝数量为约束条件,建立了混合整数线性规划模型,并采用分支界定法对模型进行求解。

可见,传统优化调度方法来源于运筹学的基本原理,能够给出最优的调度方案,但该类方法大多基于某些理想化的假设,远不能充分反映实际生产环境,距离实用还有较大距离。此外,大多数优化方法诸如动态规划算法与分枝定界算法等等是建立在对可能调度的部分枚举上,且计算复杂,因此只适合于小规模调度问题。

2) 智能优化调度

智能优化调度方法主要包括智能搜索算法和神经网络。R. Rai^[94]研究了与Atmani A^[69]相同的问题,即针对FMS中机床-刀具选择和操作分派的决策问题,以机械加工时间最小化,物料搬运及设置的成本最小化为目标,以加工时间、刀具库容量和刀具寿命为约束条件,建立了模糊多目标线性规划模型,该模型改变了以往按确定性目标进行优化的思路,并采用遗传算法对模型进行优化求解,采用的算例证明了模型的有效性和可行性。ZHANG Xiaoming^[95]提出一个解决刀具计划和工件刀具分派的

二阶方法,在第一阶段使用贪心算法确定总的刀具数量,第二阶段提出一种基于拍卖算法的工件和刀具的调度方法。Felix T.S^[96]研究了与[90]同样的问题,只是采用了蚁群算法对模型进行优化求解。T. Prabaharan^[97]针对带有中央刀库的FMS,对操作刀具调度问题提出了优先权规则调度算法和模拟退火算法,并以完成时间和仿真时间为指标对两种算法进行了比较,结果表明后者可以获得近优的解决方案。徐玲^[98]为预测刀具寿命,引入人工神经网络技术,建立了刀具寿命预测神经网络模型,同时对切削参数进行优化选择。在刀具寿命预测中,采用遗传算法训练反向传播神经网络,设计了进化神经网络的学习算法。为刀具需求计划制定、刀具成本核算,以及切削参数制定提供理论依据,节约了制造执行系统中的生产成本。席光辉^[86]针对柔性制造系统刀具需求模型及求解中存在的问题,提出以零件加工时间最短为优化目标,以刀具购买资金为约束,基于遗传法求解的新型柔性制造系统(FMS)刀具需求规划模型。通过比较分析,验证了该FMS刀具需求规划模型及其遗传算法求解需求问题的合理性和优越性。

可见,智能局域搜索法由于具有普遍适用性和较低的经验复杂性等优点,得到了广泛的重视和应用。

综上所述,现有的刀具流优化调度研究主要针对刀具管理的优化,也即如何更好地维护系统中的各类刀具,如编码管理,查询管理,以及刀具数量控制等等,而与系统实时控制相关的刀具分派优化研究,特别是从工件流和刀具流的相互作用关系出发进行有效的优化调度研究还很少见。

1.2.4 刀具流无死锁调度方法研究综述

随着柔性制造系统复杂性的增加,死锁现象的分析越来越受到研究人员的重视。死锁是在制造自动化系统中由于并发的加工过程共享有限的资源(机床、刀具及物料传输系统等)而引起的工件流动受阻,加工无法继续进行的现象。死锁现象一旦发生,除非引入外界及时的控制,否则部分甚至整个自动化制造系统将陷入瘫痪或停滞状态,从而严重影响到自动化生产,因此在设计FMS时必须重点回避死锁现象的发生。

根据分析对象的不同,死锁主要包括工件流死锁和刀具流死锁两个方面,它们都是FMS实时控制的重点和难点。两类死锁的形成机理和表现形式有所不同,应分别加以分析研究。长期以来学者们重点研究了工件流死锁问题^[99-108],但刀具流死锁的研究却只在近几年才得到应有的关注。以下分别从刀具流的死锁机理及刀具流死锁的研究方法两方面进行综述。

(1) 刀具流的死锁机理

在工件流问题中,工件流死锁表现为若干台机床之间发生循环等待现象,也即形成一个机床等待集合,集合中任一机床当前加工工件都在等待集合中另一台机床的加工,这种现象的发生是由于系统中有限的机床设备缓冲区容量,以及不恰当的工件调

度策略所引发的。显然，工件流死锁来源于缓冲区资源的有限性^[64,109-112]。

在刀具流问题中，由于系统中刀具资源的有限性，以及不恰当的刀具调度策略，就会导致刀具流死锁，即形成一个机床等待集合，集合中任意一台机床都在等待集合中其它机床释放刀具，也即若干台机床之间发生循环等刀。根据服务对象的不同，可以将刀具流死锁大体上分为两类：即基于工序备刀前提下的死锁^[62,64]和基于工步备刀前提下的死锁^[66]。

工序备刀前提是指工件在加工中心接受加工处理前，加工机床就需要从刀具流中获取刀具，每次准备刀具的数目为工件在本次装夹后，在该加工机床上接受加工所需的刀具总和，直到所有工件加工所需的刀具类型就绪后，机床才开始工作。加工完毕，加工机床释放被占用的刀具。此处以两个机床循环等刀为例说明其死锁机理，如图 1.2(a)所示。

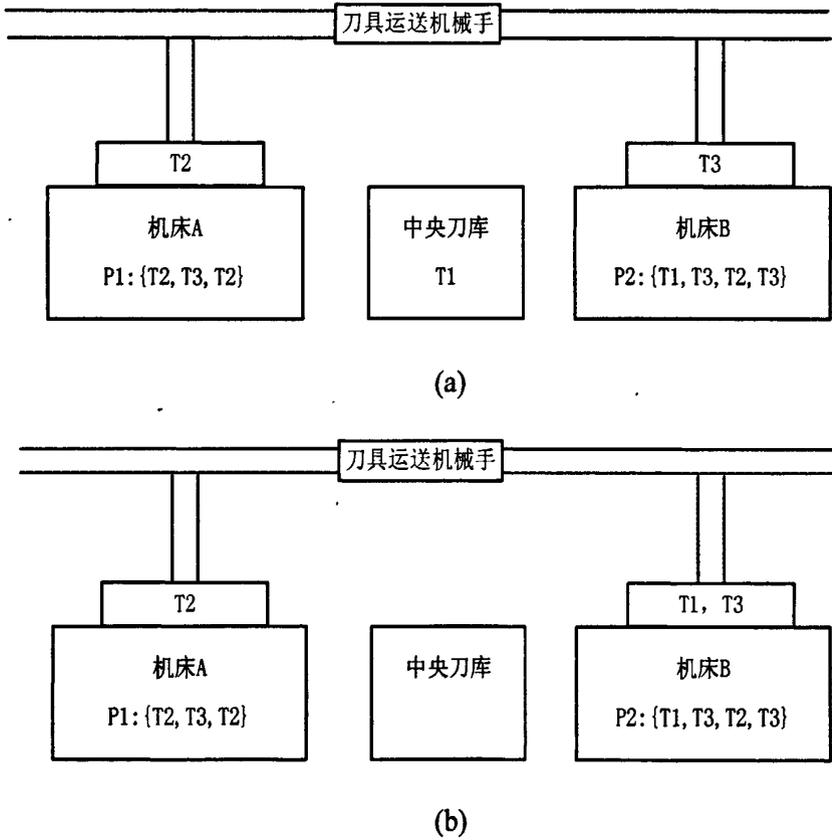


图 1.2 工序备刀下的刀具流死锁示意图

Fig.1.2 Schematic diagram of tool flow deadlock in case of procedure provisioning tools

- 工件 P1 在机床 A 上接受加工所需要的刀具集合： $\{T_2, T_3, T_2\}$ ；
- 工件 P2 在机床 B 上接受加工所需要的刀具集合： $\{T_1, T_3, T_2, T_3\}$ ；
- T_1 、 T_2 和 T_3 三类刀具各有一把；

- 刀具分派规则：对那些不需要运输的刀具给予优先权；
- 因此分派 T_2 类刀具实例给工件 P1，分派 T_3 类刀具实例给工件 P2，如图 1.2(b) 所示；
- P1 需要申请到 T_3 之后才能开始加工，P2 需要申请到 T_1 和 T_2 之后才能开始加工，故中央刀库的 T_1 类刀具分派给 P2；
- 产生死锁状态：P1 占有 T_2 申请 T_3 ，P2 占有 T_1 和 T_3 等待 T_2 。

工步备刀前提是指工件在加工中心接受加工处理前，加工机床就需要从刀具流中获取刀具，在获取工件在该加工机床上接受加工所需要的第一把刀具后，即可开始加工，在加工过程中，为减少机床的空闲时间，应使刀具流准备刀具时间与当前工步的加工时间充分重叠，刀具流应为该工件的下一工步准备刀具。当前工步加工完毕，若工件的后续工步还需要此刀具，则占用此把刀具，否则释放该刀具。以两个机床循环等刀为例，说明其死锁机理，如图 1.3(a) 所示。

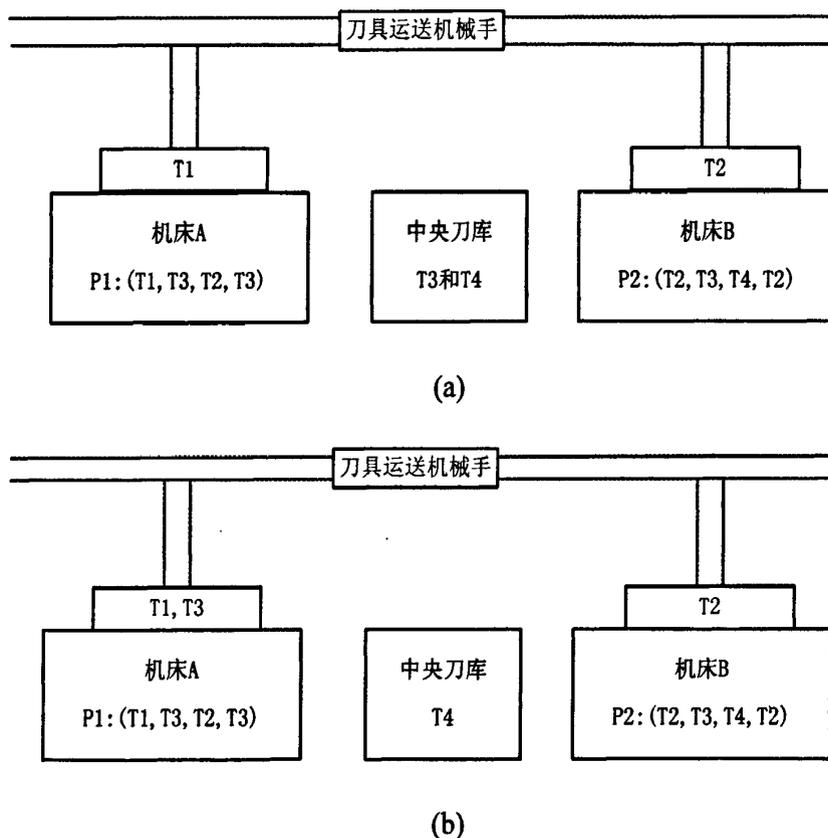


图 1.3 工步备刀下的刀具流死锁示意图

Fig.1.3 Schematic diagram of tool flow deadlock in case of working step provisioning tools

- 工件 P1 在机床 A 上接受加工所需要的刀具顺序：(T_1, T_3, T_2, T_3)；
- 工件 P2 在机床 B 上接受加工所需要的刀具顺序：(T_2, T_3, T_4, T_2)；

- T_1 、 T_2 、 T_3 和 T_4 四类刀具各有一把；
- 刀具分派规则：对那些不需要运输的刀具给予优先权；
- 状态 1：P1 用刀具 T_1 加工，为其下一工步申请刀具 T_3 ；P2 用刀具 T_2 加工，为其下一工步申请刀具 T_3 ；
- 将中央刀库中的刀具 T_3 按优先权规则分派给 P1；如图 1.3(b)所示；
- 状态 2：P1 用刀具 T_3 加工，为其下一工步申请刀具 T_2 ；P2 用刀具 T_2 加工，为其下一工步申请刀具 T_3 ；（刀具被占有是因为工件的后续工步还需要它们）。
- 死锁状态发生：P1 占用刀具 T_3 申请刀具 T_2 ；P2 占用刀具 T_2 申请刀具 T_3 。

根据以上描述，不难看出刀具流发生死锁必须具备四个必要条件^[113-114]：

- 1) 互斥：某一刀具资源只能分配给某个确定的机床，不能同时为两个机床占据；
- 2) 非抢占：刀具资源不可抢占，只能根据占用它的机床自愿释放；
- 3) 占有并等待：占据某些刀具资源的机床，请求另外新的刀具资源，而这些刀具资源为其它机床占有并等待。
- 4) 循环等待：存在一组刀具资源请求 $\{M_1, M_2, \dots, M_n\}$ ，其中 M_1 等待 M_2 占据的刀具资源， M_2 等待 M_3 占据的刀具资源， M_n 等待 M_1 占据的刀具资源。

(2) 刀具流死锁的研究方法

虽然刀具流死锁问题与工件流死锁有本质区别，但是工件流死锁研究中的一些死锁处理方法仍然有良好的借鉴作用。综合已有的工件流死锁研究的思想，可以发现工件流死锁研究中处理死锁的策略主要包括：死锁预防，死锁检测与恢复和死锁避免。

1) 死锁预防^[115-118]：基本思想是从控制形成死锁的必要条件出发，确保死锁的必要条件不能同时满足。该方法对任务和资源之间的交互施加限制，以使导致死锁的请求不会被满足。预防机制的优点在于，它不需要知道系统的状态来实现控制，只需要一个简单的控制法则，但该方法不能保证所获得的解的性能，在不同运行状态下的解的性能差异可能很大。

2) 死锁检测和恢复^[119-121]：检测恢复方法利用监控机制来检测死锁。死锁检测模块始终监听系统实时状态，一旦检测到死锁形成，死锁恢复模块将被启动，通过结束某些死锁的任务以从死锁状态中恢复过来。此种策略通常比一般的预防方法能够提供更高的资源利用率。一般用在死锁很少出现，检测和恢复过程费用不高，并且动作过程要比死锁的产生要快的情况下。

3) 死锁避免^[122-127]：通过记录系统的当前状态和可能的未来条件，来使得死锁的一个或多个条件为假。这是一种动态的方法，它能够利用当前刀具资源分配的情况和进程的未来行为，来控制刀具资源的释放和获取。死锁避免策略的主要特点是它是实时工作的，决策的依据是系统资源当前的状态信息。

类似的，刀具流中解决死锁引用了与工件流死锁相同的处理方法，相应的研究死

锁的工具具有 Petri 网和图论两种。

目前对刀具流死锁控制分析的研究不多,一些学者正在致力于这方面的工作。GEBRAEEL^[66]较为系统地研究了刀具共享系统中死锁的检测、预防和避免问题。首先基于图理论给出了刀具流死锁定义并证明了最优死锁调度问题是 NP-hard 问题,然后讨论了用于死锁避免的银行家算法和用于死锁预防的顺序分派结构,并证明这些方法确实可使系统免于陷入死锁状态,但同时也证明了它们只是一种次优的方法,因为它们限制了刀具的分派种类,尤其是优先权高的工件也不能保存刀具为将来的使用,但是该文在分析过程中没有考虑刀具寿命对死锁的影响。毕诸明^[62]为预测被测 FMS 运控软件中刀具流引起的死锁和回避功能。首先建立了刀具流的 PN 模型,然后从网模型中变迁的等待关系来定义 FMS 刀具流的死锁现象,并给出了刀具流死锁检测算法,但是该 PN 模型只局限于刀具流内部,没有考虑工件流对刀具流的影响。舒海生^[64]充分考虑了工件流对刀具流的制约作用,在刀具流调度部分首先提出了刀具预准备方法,针对工序备刀情况进行了死锁分析,采用图论方法定义了刀具申请分配图(TAAG),并将刀具流死锁与 TAAG 联系起来,建立了刀具流死锁的相关理论。然而 FMS 的实际运行中,各类刀具都有不同寿命,并且还可能出现各种不可预期的失效(如:刀具的磨损和破损),在刀具失效情况下如何进行刀具流死锁性质的分析,该文没有提及。

综上所述,从三个死锁处理策略的机理来看,死锁检测/恢复策略要求不断地检测死锁,发生死锁之后的死锁恢复过程将不可避免地使得原有顺畅的生产过程被打乱并重新调整,从而降低了生产率。死锁预防策略又在很大程度上限制了生产的灵活性。相比较而言,死锁避免策略是比较有效的避免死锁的方法,它可以利用当前的状态信息和加工流程控制资源的分配,特别是与有效的启发式算法相结合将可以实现无死锁条件下的优化调度。从死锁研究的工具来看:随着制造系统中工件和资源的数量、种类的增加,采用 Petri 网工具进行分析将引起系统控制模型的复杂化,不利于构建高效的实时控制算法,而图论工具能够在较高的层次上对刀具流死锁问题加以抽象,隐藏了那些与问题实质无关的次要因素,使得分析域更加清晰,从而更有利于复杂系统的死锁问题分析。

1.3 研究目的及意义

刀具流是柔性制造系统(FMS)中的一个重要组成部分,在 FMS 的生产过程中占有十分重要的地位。它的调度是否合理,管理是否完善,控制是否有效,对整个柔性制造系统的利用率、生产效率、生产质量和加工成本都将产生很大的影响。当前刀具流调度理论的薄弱已经导致了实际生产中 FMS 运行困难,阻碍了 FMS 生产能力的进一步发挥。因此,为了提高系统的总体加工性能和充分发挥 FMS 本身的柔性,进行更

深入的刀具流规划与调度研究具有重要的理论价值和实际意义。

本课题的研究目的旨在建立和完善刀具流调度理论和相关关键技术，针对 FMS 运行中刀具种类繁多、刀具更换频繁和刀具资源有限等特点，分别进行基于“仿真+规则”的启发式调度研究和基于智能搜索算法的智能优化调度研究，建立适用的刀具流调度策略和算法，为充分发挥系统生产能力，提高系统生产效率提供可靠的技术保障。

1.4 论文的总体框架和研究内容

本文在现有的国内外研究基础上，综合应用面向对象技术、Petri 网方法、图论方法及智能优化算法等，深入研究 FMS 刀具流调度的一系列关键问题，总体研究框架如图 1.4 所示。首先，针对现有 FMS 建模研究的不足，建立更为全面的 FMS 调度仿真模型，为后续的刀具流调度研究提供指导。其次，针对现有刀具流调度研究的不足，将刀具寿命考虑进来，对工序备刀情况下的启发式调度作进一步研究，然后重点对工步备刀情况下的启发式调度与最优化调度进行深入分析。最后，对上述启发式调度算法和优化调度算法的有效性进行仿真对比与验证并作出综合评价。

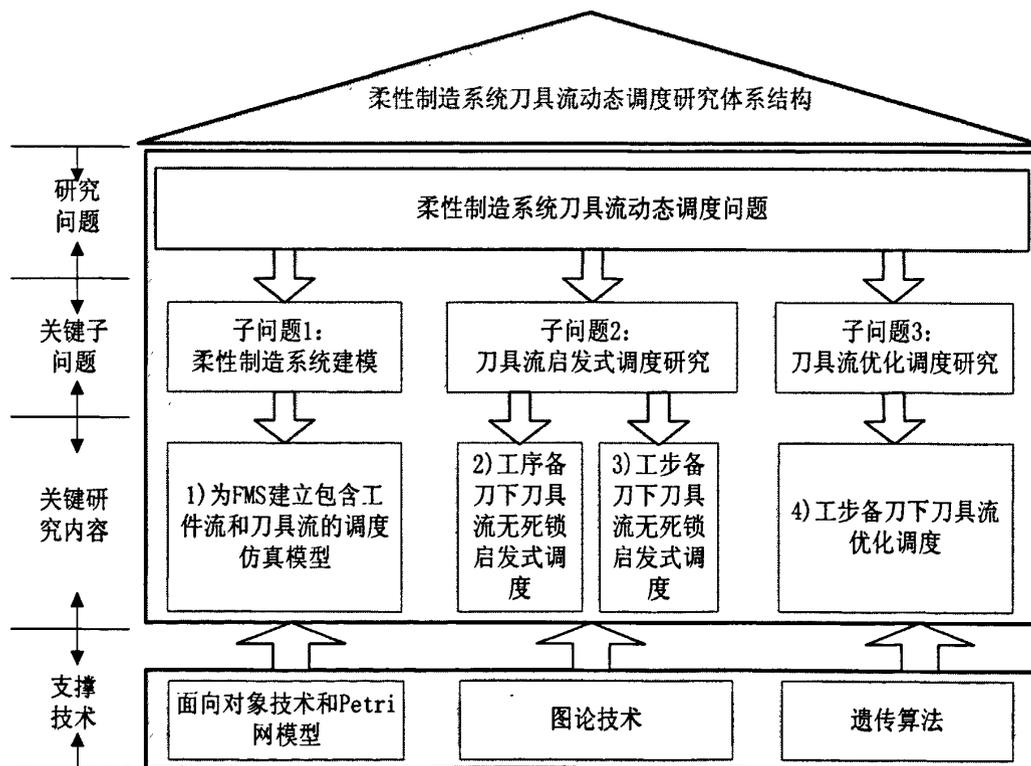


图 1.4 论文的总体研究框架

Fig.1.4 Overall research contents framework of dissertation

本文围绕 FMS 中的刀具流调度，针对调度过程中涉及的刀具流建模、刀具流死

锁避免、刀具的实时分派等方面做以下的研究工作:

(1) 结合面向对象技术和 Petri 网理论分别为 FMS 生产中的工序备刀情况和工步备刀情况建立仿真模型, 该模型全面包括工件流和刀具流进程, 并将机床故障与刀具失效作为一个组成部分包括进来, 为后续的刀具流启发式调度研究提供指导。

(2) 针对工序备刀前提下的刀具流无死锁调度进行研究。建立考虑刀具寿命的刀具申请分配图, 并将其和刀具流死锁联系起来, 在此基础上深入分析考虑刀具实际寿命的刀具流死锁的相关性质, 建立刀具流死锁判定的相关理论, 提出刀具流死锁避免策略, 并通过实验验证两层死锁避免策略的合理性和可行性。构建刀具分派的启发式调度规则, 给出刀具流无死锁的启发式调度算法, 解决工序备刀下的刀具流死锁避免和刀具实时分派问题。

(3) 探讨工步备刀前提下刀具流的无死锁调度问题。针对为工步备刀的刀具流建立刀具流状态演变模型和刀具资源分派图, 给出刀具流死锁状态和无死锁不安全状态的定义, 并建立相应的检测算法, 在此基础上给出刀具流的死锁避免算法。通过实例验证死锁避免算法的合理性和可行性。建立工步备刀下刀具流无死锁的启发式调度算法并对动态调度过程中的刀具失效调度给出相应的处理策略, 解决工步备刀下的刀具流死锁避免和刀具实时分派问题。

(4) 研究工步备刀前提下刀具流的优化调度问题。针对工步备刀下的刀具流建立数学模型, 并提出一种双重遗传算法对模型进行优化求解, 通过实例验证双重遗传算法的有效性和可行性。进而针对系统中的刀具破损等突发事件, 构建基于双重遗传算法的准静态再调度方法, 解决并优化工步备刀下的刀具实时分派问题。

(5) 评价刀具流的各调度策略和算法。对各调度算法在有无故障和无故障两种情况下进行仿真实验。通过对实验结果的分析来验证各调度算法和调度策略的可行性和有效性, 最后对它们做出综合评价。

第 2 章 工件流和刀具流的 Petri 网模型

FMS 是一个复杂的系统,对其合理的建模是后续调度控制的必要条件。在众多的建模方法中, Petri 网以其坚实的数学基础和强大的描述能力,为广大学者所采用。本章也以 Petri 网为建模工具,结合面向对象技术和分层递阶思想分别针对工序可中断和不可中断两种前提为工件流和刀具流进行建模。通过对系统中的各类设备资源进行划分,将纷繁复杂的资源分为有限种对象类别。针对每一类对象,根据其可能的状态和动作将其以 Petri 网形式加以描述,形成对象子网。在此基础上,通过消息和消息变迁进一步将各个对象子网联成为一个完整的 Petri 网模型。

2.1 FMS 的建模过程

由于 FMS 非常复杂,所以对其建模时,若直接以系统中的加工设备资源为个体进行建模,会导致系统模型过于庞大,给系统分析带来困难。故在建模时采用有色 Petri 网与面向对象技术融合的思想,即在建模方法上采用面向对象技术,把系统中的各类设备资源分割成若干个对象实体,通过消息和消息变迁实现各对象实体之间的通信。在对象的具体实现上则根据每类对象实体可能的状态和动作将其以 Petri 网形式加以描述。

基于上述有色 Petri 网与面向对象技术融合的思想, FMS 整个建模过程如图 2.1 所示,主要分为三个阶段:系统的静态分析及对象划分,设备资源对象类子网建模和各设备资源对象子网的集成。

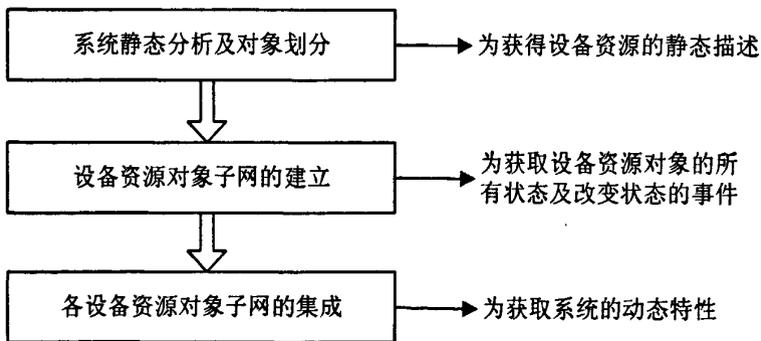


图 2.1 FMS 的建模过程

Fig.2.1 Modeling procedure of FMS

对 FMS 进行建模时,首先要对系统中的各设备资源进行对象划分,并对各个设备资源对象进行静态描述,也即建立设备资源对象模型,在此基础上,确定各类设备资源对象所有可能的状态及改变状态的动作(或事件),也即建立各设备资源对象子

网，但是孤立的设备资源对象子网并不能帮助我们了解整个系统的动态特性，因此，必须通过信息传递机制实现各孤立对象子网的通信，也即对各设备资源对象子网进行集成，从而建立工件流和刀具流 Petri 网模型。

2.2 工件流和刀具流 Petri 网的实现

2.2.1 设备资源类对象划分及描述

设备资源描述是所有模型的基础，而 FMS 中的实体设备资源之间的逻辑关系非常复杂，若直接以各加工设备资源为个体进行建模，会导致系统模型过于庞大，不利于系统的分析。因此利用面向对象的方法，将 FMS 中的所有设备资源抽象为设备基类，派生出 5 类对象：工件类，机床类，工件小车类，刀具机械手类和刀具类。设备资源对象类与柔性制造系统中真实的制造资源设备相对应，可准确的抽象出对应设备的属性和活动。它们的属性包括名称、编号、位置、状态（如忙态、闲态和故障态）、统计（如生产率，设备利用率）等信息。活动包括初始化、设置设备状态、计算利用率及执行调度规则等。采用 Coad / Yourdon 方法表示的设备资源对象模型如图 2.2 所示。

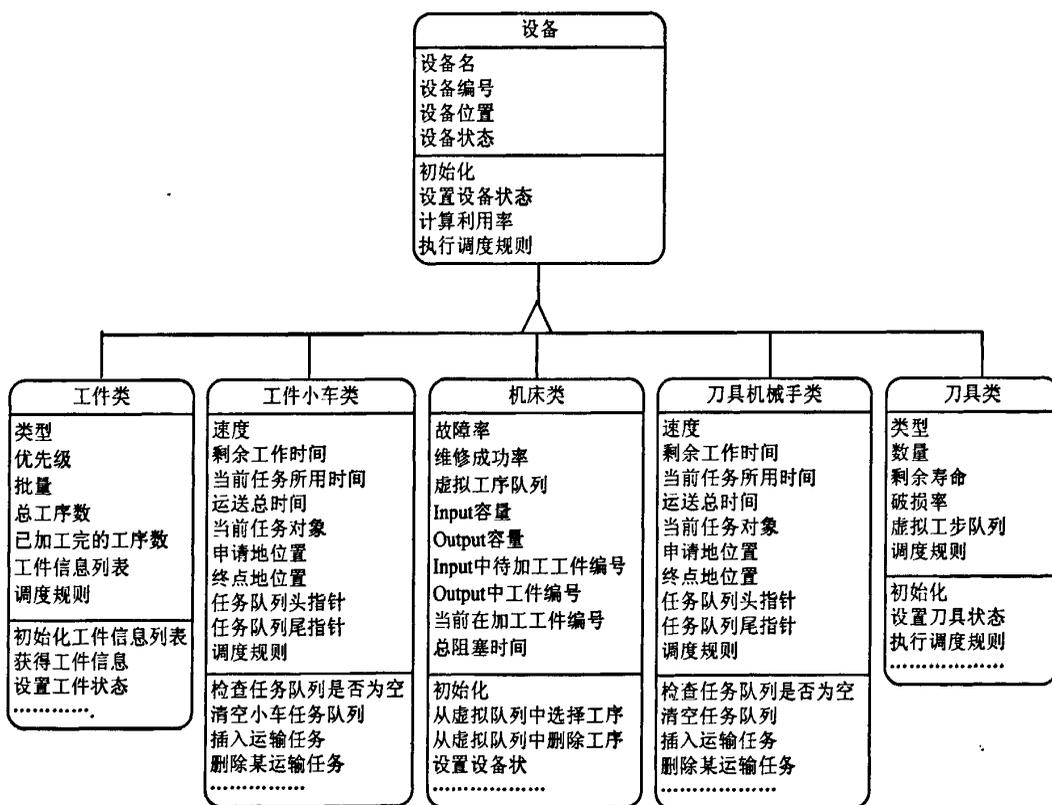


图 2.2 设备资源对象模型

Fig.2.2 Modeling of device resources as object

2.2.2 设备资源对象子网建模

具有相似状态和动作的加工设备实体被封装成一个设备资源对象子网，且各设备资源对象子网是独立的。由于基本 Petri 网中的托肯，只能表示一种资源，而在制造系统中，资源是多种的，有机床、工件、AGV(工件小车)、刀具机械手和刀具等，为了使托肯表示系统中多种资源的组合，采用着色 Petri 网建立设备资源对象子网。首先给出其形式化定义。

定义 1 $OP_i^N = (P_i, T_i, I_i, O_i, IM_i, OM_i, C_i, M_{i,0})$

其中：

OP_i^N 是对象子网 i 的 Petri 网模型；

$P_i = \{p_1, p_2, \dots, p_m\}$ 是有限的位罝（状态）集合；

$T_i = \{t_1, t_2, \dots, t_m\}$ 是有限的变迁（操作）集合；

$I_i: P \times T \rightarrow N$ 是由位罝到变迁的输入函数；

$O_i: T \times P \rightarrow N$ 是由变迁到位罝的输出函数；

IM_i 为对象 OP_i^N 的接收消息库所；

OM_i 为对象 OP_i^N 的发送消息库所；

$C(P_i)$ 为对象 OP_i^N 所有状态库所的颜色集；

$C(T_i)$ 为对象 OP_i^N 所有变迁的颜色集；

$C(IM_i)$ 为对象 OP_i^N 所有输入消息库所的颜色集；

$C(OM_i)$ 为对象 OP_i^N 所有输出消息库所的颜色集；

$M_{i,0} = \{MM_{i,0} \cup SM_{i,0}\}$ 为 OP_i^N 的所有初始标识集合，其中 $MM_{i,0}$ 为 OP_i^N 的所有消息库所的初始标识， $SM_{i,0}$ 为 OP_i^N 的所有状态库所的初始标识。

(1) 工件对象子网

工件对象子网的 Petri 网模型如图 2.3 所示。 p_1^1 ：工件等待加工态； t_1^1 ：请求加工； p_1^2 ：工件正在加工态； t_1^2 ：某工序完成； t_1^3 ：某工序中断； OM_1^3 ：工件发出请求机床消息； IM_1^{3A} ：接收机床发出的某工序完成消息； IM_1^{3B} ：接收机床发出的某工序中断消息。

对于工件对象，其最初始状态为“等待加工态”，故其向机床对象发送请求加工消息，申请其进入当前加工机床的输入缓冲区，若工件被其加工机床确定为加工对象，且机床的输入缓冲区有空闲位罝，则通过工件小车 AGV 的运输，将工件运送到机床的输入缓冲区，当机床对象处于闲态时，则将工件搬上机床，工件转向“正在加工态”。若机床在加工工件时突然发生故障，则工件对象将会接收到一个机床对象发出的“工序加工中断”消息，变迁“工序中断加工”发生，工件重新转向“等待加工态”。若机床加工完毕，且工件仍有待加工工序，则工件对象接收到机床对象发出的“工序加

工完毕”消息，工件重新转向待加工工序的“等待加工态”。

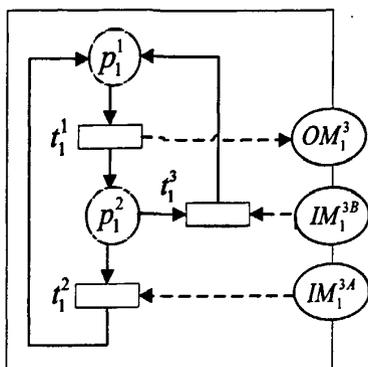


图 2.3 工件对象子网

Fig.2.3 Subnet of parts as object

(2) 工件小车对象子网

AGV 对象子网的 petri 网模型如图 2.4 所示。 p_2^1 : 小车闲态; t_2^1 : 小车被申请; p_2^2 : 小车预定态; t_2^2 : 工件开始运送; p_2^3 : 小车忙态; t_2^3 : 工件运送到达; IM_2^3 : 工件小车类接收机床类发出的请求小车消息; OM_2^3 : 小车类向机床类发出工件运送到达消息。

对于工件小车对象，当小车处于闲态且接收了机床发出的请求小车消息时，小车转为预定态，当小车开始运送时，状态转为忙态，当工件运动到达时，工件小车对象可以发送一个“运送到达”消息给机床对象同时转为闲态。

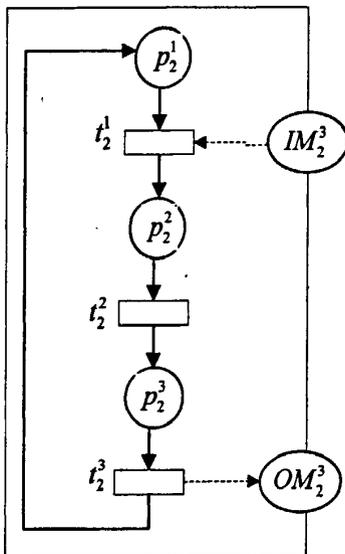


图 2.4 AGV 对象子网

Fig.2.4 Subnet of AGVs as object

(3) 机床对象子网

1) 工序备刀情况下，机床对象子网的 Petri 网模型如图 2.5 所示。

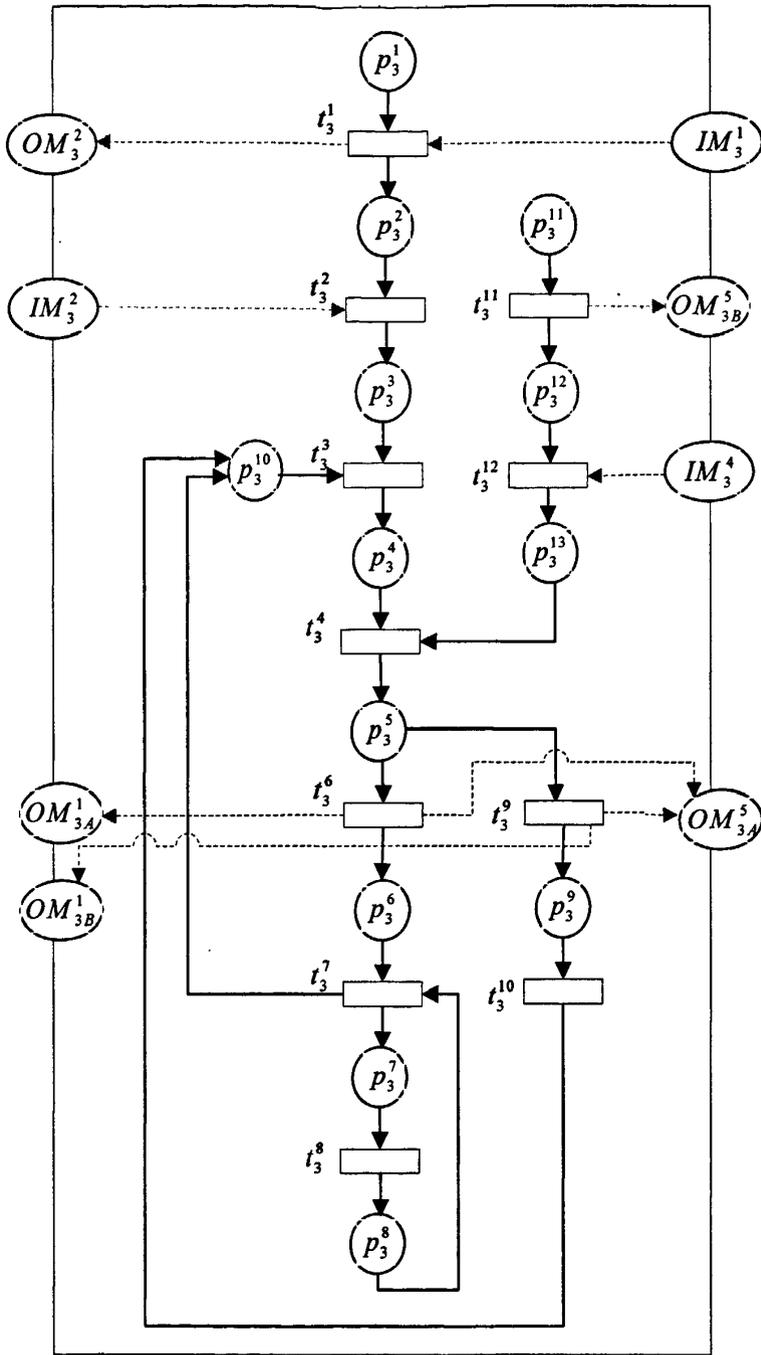


图 2.5 工序备刀下机床对象子网

Fig.2.5 Subnet of machines as object in case of procedure provisioning tools

p_3^1 : 输入缓冲区有空位; t_3^1 : 输入缓冲区被预定并请求工件小车 AGV; p_3^2 : 输入缓冲区预定态; t_3^2 : 输入缓冲区被工件占用; p_3^3 : 输入缓冲区有工件; t_3^3 : 工件被搬上机床; p_3^4 : 机床预定态; t_3^4 : 机床开始加工; p_3^5 : 机床忙; t_3^6 : 工序加工完毕; p_3^6 : 机床阻塞态; t_3^7 : 工件进入输出缓冲区; p_3^7 : 输出缓冲区有工件; t_3^8 : 工件被运

走; p_3^8 : 输出缓冲区有空位; t_3^9 : 机床发生故障; p_3^9 : 机床故障态; t_3^{10} : 维修完毕; p_3^{10} : 机床闲态; p_3^{11} : 当前工序缺刀态; t_3^{11} : 申请刀具; p_3^{12} : 当前工序等刀态; t_3^{12} : 刀具到达; p_3^{13} : 当前工序所需刀具就绪态。 OM_1^1 : 机床对象接收工件对象发出的请求机床消息; OM_3^2 : 机床对象向工件小车对象发出的请求小车消息; IM_3^2 : 工件小车对象向机床对象发出的工件运送到达消息; OM_{3A}^3 : 机床对象向工件对象发出的某工序完成消息; OM_{3B}^4 : 机床对象向工件对象发出的某工序中断消息; OM_{3B}^5 : 机床对象向刀具对象发出的请求刀具消息; OM_{3A}^6 : 机床对象向刀具对象发出的释放刀具消息; OM_3^4 : 运刀机械手对象向机床对象发出的刀具运送到达消息。

机床对象接收到工件对象的加工请求后, 根据其输入缓冲区的状态对工件的加工请求消息给予响应, 若机床的输入缓冲区有空闲位置, 则向工件小车 AGV 发出运送工件请求, AGV 为闲状态时, 先确定工件的位置然后将工件运送到机床的输入缓冲区, 如果机床对象处于闲状态, 则将该工件搬上机床, 并将机床转为被工件预定状态。机床是否开始加工则需要考虑刀具流的影响, 也即机床能否进行下一工序的加工还应判断该工序加工所需的刀具是否全部在机床刀具库中, 也即只有机床对象处于“当前工序所需刀具就绪态”才能开始下一工序的加工。机床对象处于“当前工序缺刀态”时, 将向刀具对象发出“请求刀具消息”。当机床在加工过程中, 发生故障时, 机床由忙态转为故障态, 同时向刀具对象发出释放刀具消息, 待故障维修完毕, 机床重新转为闲态。当机床工序加工完毕时, 机床即重新转为闲态, 并向工件对象和刀具对象分别发出工序加工完成消息和释放刀具消息。

2) 工步备刀情况下, 机床对象子网的 Petri 网模型如图 2.6 所示。

p_3^1 : 输入缓冲区有空位; t_3^1 : 输入缓冲区被预定并请求 AGV; p_3^2 : 预定态; t_3^2 : 输入缓冲区被工件占用; p_3^3 : 输入缓冲区有工件; t_3^3 : 工件被搬上机床; p_3^4 : 机床预定态; t_3^4 : 机床开始加工; p_3^5 : 机床忙; t_3^6 : 工序加工完毕; p_3^6 : 机床阻塞态; t_3^7 : 工件进入缓冲区; p_3^7 : 输出缓冲区有工件; t_3^8 : 工件被运走; p_3^8 : 输出缓冲区有空位; t_3^9 : 工步加工完成; t_3^9 : 机床发生故障; p_3^9 : 机床故障态; t_3^{10} : 维修完毕; p_3^{10} : 机床闲态; p_3^{11} : 当前工步缺刀态; t_3^{11} : 申请刀具; p_3^{12} : 当前工步等刀态; t_3^{12} : 刀具到达; p_3^{13} : 当前工步所需刀具就绪态。 OM_3^2 : 机床类向工件小车类发出的请求小车消息; IM_3^2 : 小车类向机床类发出的工件运送到达消息; OM_{3A}^3 : 机床类向工件类发出的某工序完成消息; OM_{3B}^4 : 机床类向工件类发出的某工序中断消息; OM_{3B}^5 : 机床类向刀具类发出的请求刀具消息; OM_{3A}^6 : 机床类向刀具类发出的释放刀具消息; IM_3^4 : 刀具机械手类向机床类发出的刀具运送到达消息。

机床对象接收到工件对象的加工请求后, 根据其输入缓冲区的状态对工件的加工请求消息给予响应, 若机床的输入缓冲区有空闲位置, 则向工件小车 AGV 发出运送工件请求, AGV 为闲状态时, 先确定工件的位置然后将工件运送到机床的输入缓冲

区, 如果机床对象处于闲状态, 则将该工件搬上机床, 并将机床转为被工件预定状态。机床能否开始下一步的加工还应取决于对当前工步加工所需刀具是否在机床刀具库中, 只有当机床对象处于“当前工步所需刀具就绪态”方能正常进行加工。当机床的当前工步加工完成时, 机床重新转为预定态, 并在下一工步所需刀具就绪时开始加工。当工序加工完成时, 机床重新转为闲状态, 并向工件类和刀具类分别发出工序加工完成消息和释放刀具消息。而对机床故障采用跟工序备刀相同的处理方式。

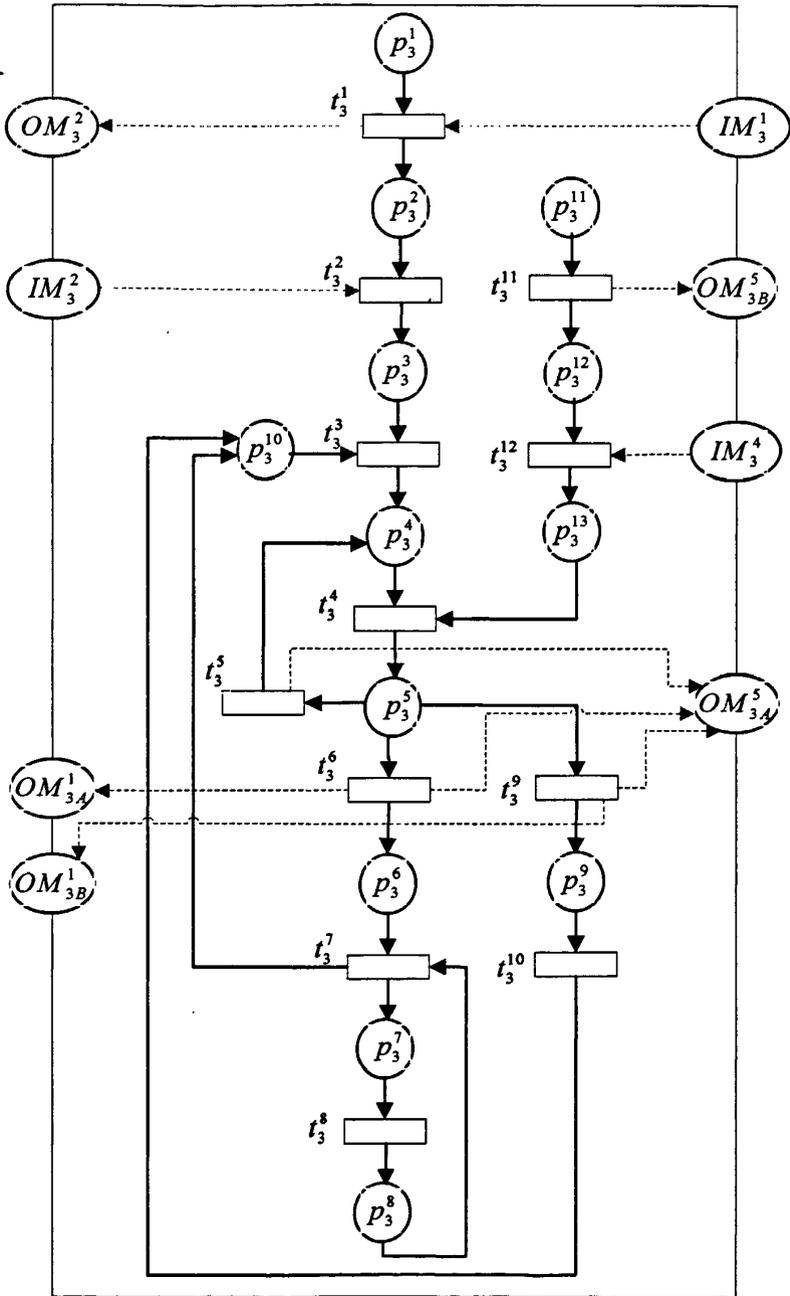


图 2.6 工步备刀下机床对象子网

Fig.2.6 Subnet of machines as object in case of working step provisioning tools

(4) 刀具机械手对象子网

刀具机械手对象子网如图 2.7 所示。 p_4^1 : 机械手闲态; t_4^1 : 机械手被申请; p_4^2 : 机械手预定态; t_4^2 : 刀具开始运送; p_4^3 : 机械手忙态; t_4^3 : 刀具运送到达; IM_4^3 : 刀具机械手接收刀具发出的请求机械手消息; OM_4^3 : 刀具机械手类向机床类发出刀具运送到达消息。

对于刀具机械手对象, 当机械手处于闲态且接收了刀具发出的请求机械手消息时, 机械手转为预定态, 当机械手开始运送刀具时, 由预定态转为忙态。当刀具运送到达时, 刀具机械手对象可以发送一个“刀具运送到达”消息给机床对象同时转为闲态。

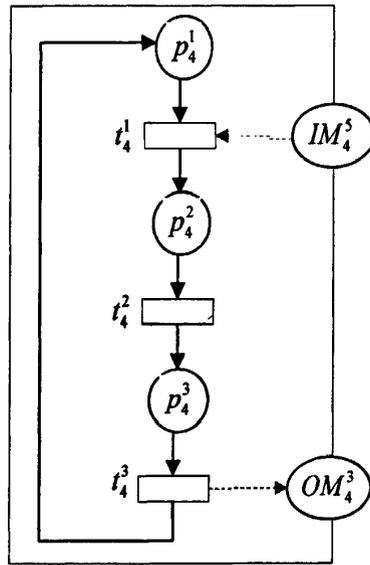


图 2.7 刀具机械手对象子网

Fig.2.7 Subnet of carrier manipulators as object

(5) 刀具对象子网

刀具对象子网的 petri 网模型如图 2.8 所示。 p_5^1 : 刀具闲态; t_5^1 : 刀具被申请; p_5^2 : 刀具预定态; t_5^2 : 开始运送; p_5^3 : 刀具忙态; t_5^3 : 刀具被释放; t_5^4 : 刀具失效; p_5^4 : 刀具故障态; t_5^5 : 刀具被运走; p_5^5 : 刀具维修态; OM_5^4 : 刀具类向机械手类发送的请求刀具机械手消息; IM_5^3A : 刀具类接收机床类发出的释放刀具消息; IM_5^{3B} : 刀具类接收机床类发出的请求刀具消息。

刀具对象涵盖了系统中所有可调动的刀具资源, 刀具对象初始状态为闲态, 当接受到机床对象发出的请求刀具消息时, 刀具转为预定态, 同时向刀具机械手对象发送一个请求运送消息, 当刀具被运送到机床刀库并开始加工时, 刀具由预定态转为忙态, 当加工过程中刀具发生磨损或破损时, 变迁“刀具失效”被激发, 刀具由忙态转为故障态, 当收到机床对象发出的释放刀具消息时, 失效刀具被运走并转为维修态, 也即失效刀具被运送到中央刀库的故障缓冲区等候处理。

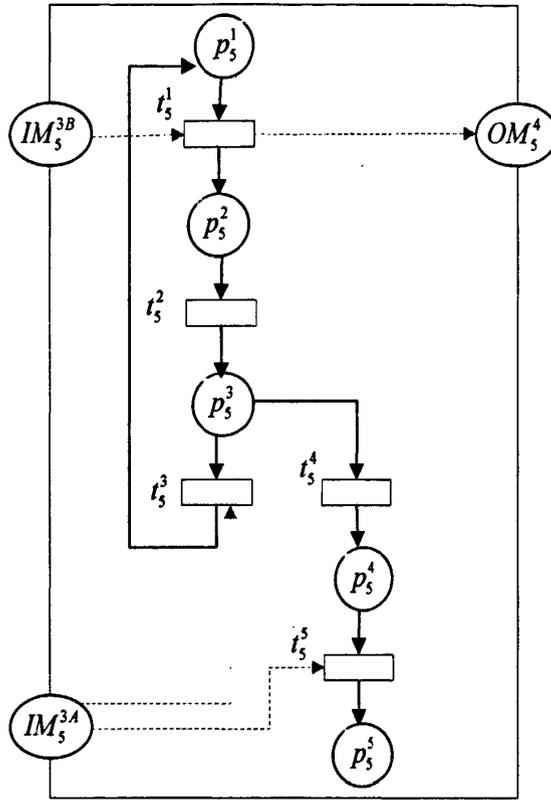


图 2.8 刀具对象子网

Fig.2.8 Subnet of tools as object

2.2.3 工件流和刀具流的 Petri 网模型

(1) 设备资源对象子网的集成

集成后的设备资源对象子网可以描述为系统内若干的设备资源对象及它们之间的消息传递关系，下面给出其形式化定义。

定义 2 $R_j = (OM_i, G_j, IM_j)$

其中：

R_j 为发送消息对象 OP_i^N 与接收消息对象 OP_j^N 间的通信关系

OM_i ：对象子网 OP_i^N 的发送消息库所；

IM_j ：对象子网 OP_j^N 的接收消息库所；

$G_j = \{g_k | k=1,2,\dots\}$ 是位于 OP_i^N 与 OP_j^N 之间的一类特殊变迁，称为消息变迁。

(2) 基于分层递阶思想的工件流和刀具流的 Petri 网模型

要想准确的描述 FMS 的运行过程，必须针对其组成结构和控制方式进行建模，根据柔性制造系统的分层分布式控制体系结构特征，将工件流和刀具流 Petri 网模型分为两层：工作站层和控制决策层。其中工作站层指的是集成后的设备资源对象子网，其主要任务是接受来自控制决策层的命令，对工作站内的各设备进行规划与调度，并

实时获得工作站内各设备的状态数据，如：加工机床和输送系统等装置的运行时间、停机时间及故障原因，刀具的寿命、破损断裂情况及地址识别，零件的实际加工工位、加工时间等。最后将命令的执行情况和 workstation 内的设备状态数据信息反馈给控制决策层。控制决策层的工作任务主要是如何根据 workstation 层反馈的状态数据信息来实时地调度工件和刀具的流动，也即控制工件和刀具的分派及释放，其调度技术和水平直接影响到整个系统的生产效率及经济效益。

基于分层递阶思想和定义 2，建立工件流和刀具流的 Petri 网模型如图 2.9 所示。

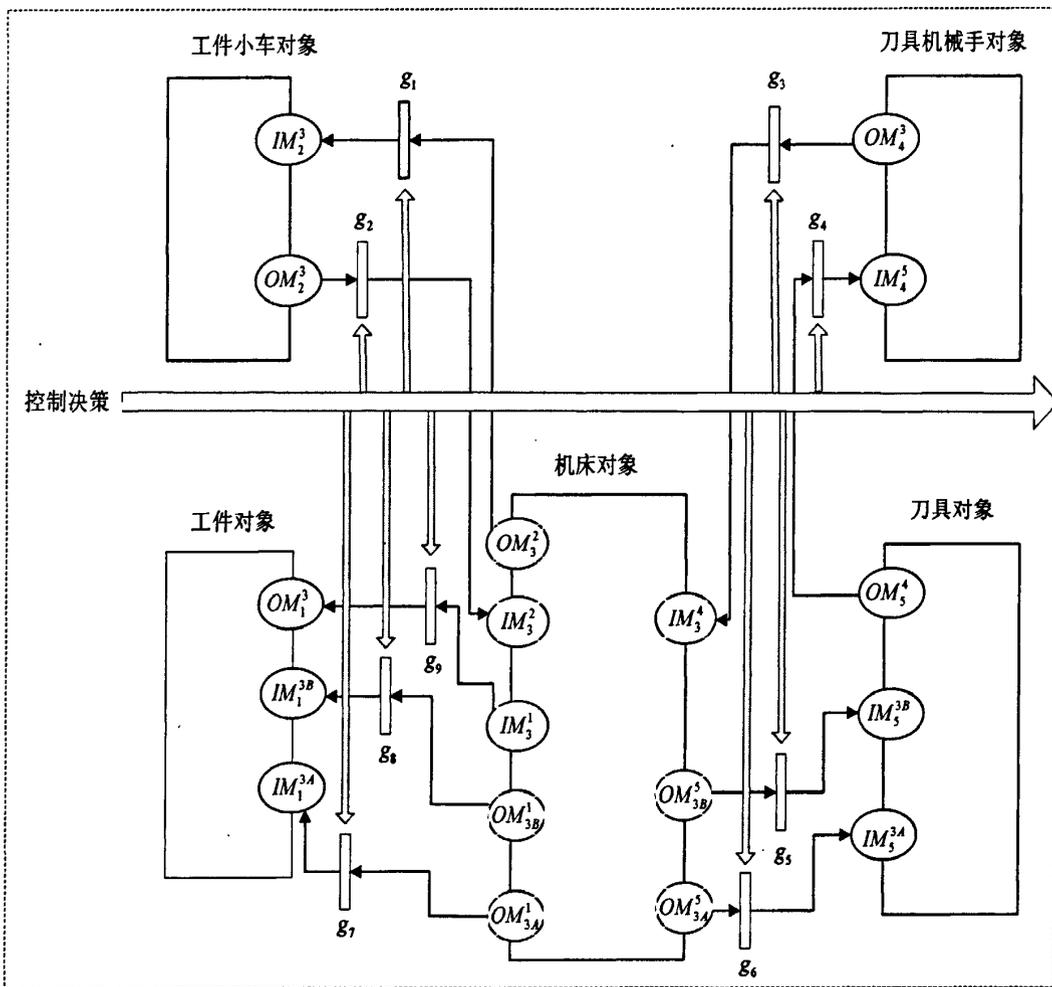


图 2.9 工件流和刀具流的 Petri 网模型

Fig.2.9 Petri net modeling of part flow and tool flow

图 2.9 中从左至右分别是工件对象子网、工件小车对象子网，机床对象子网、刀具机械手对象子网和刀具对象子网，每个对象子网中包含了该类实体资源对象所有可能的状态和动作。整个系统的运行通过激发各个对象子网之间的消息变迁来控制，并依靠引入外部的调度决策来解决模型中存在的并发和争用现象。

2.3 本章小结

结合面向对象技术和 Petri 网理论分别为 FMS 生产中的工序备刀情况和工步备刀情况建立了调度仿真模型，该模型全面包括工件流和刀具流进程，并将机床故障与刀具失效作为一个组成部分包括进来，从而准确方便地描述了考虑刀具流的 FMS 运行各个环节，为后续的刀具流启发式调度研究奠定了基础并提供了技术指导。.

第3章 基于工序备刀的刀具流无死锁调度研究

FMS 工件流和刀具流的综合 Petri 网模型从本质上高度抽象了系统的整体运行过程。在该模型中,存在着大量的 Petri 网固有的并发和争用现象,这些现象是 Petri 网本身所无法解决的,只能依靠引入有效地外部调度决策来加以处理,否则系统将无法正常运行。例如某时刻系统中可能存在若干台机床同时争用某些刀具的情况,这就是一类典型的并发争用现象。此时系统必须引入合理的外部调度来决策如何分派这些刀具,以使得系统能够进行正常的工作并获得较好的性能指标。

工件流死锁和刀具流死锁是在解决并发和争用现象时经常出现的两大类调度结果,它们的形成与调度决策的不合理有着密切的关系,但对系统的影响却是十分类似的,比如造成系统的局部运行停滞不前,严重时则会扩散到整体,使得整个系统完全陷入瘫痪。鉴于工件流死锁问题的研究已经比较深入而广泛,而刀具流死锁研究还不多见,因此本文重点针对刀具流死锁进行深入分析,最终构建刀具流无死锁调度算法。

刀具流无死锁调度是指在确保刀具流不发生死锁的前提下,为了得到较好的系统性能,如何在正确的时间为各机床分派正确的刀具。刀具流死锁避免和刀具分派是刀具流无死锁调度的两个基本问题。文献[64]中,通过定义一类刀具申请分配图(Tool Applying Allocation Graph, 简称为 TAAG),针对工序备刀情况下的刀具流死锁问题进行了理论分析,取得了一些结论,但是在研究过程中没有考虑刀具寿命对刀具流死锁的影响,而是假设每把刀具的寿命都是足够工件加工的。显然,这与实际的生产情况不相符合。为此,本章对工序备刀下考虑刀具寿命的刀具流死锁做了进一步的分析,通过建立刀具流死锁判定的相关理论,提出刀具流死锁避免策略,构建了刀具分派的启发式调度规则,进而给出了基于“死锁避免策略+启发式规则+仿真”的无死锁调度算法。

3.1 扩展的刀具申请分配图

3.1.1 刀具寿命对 TAAG 的影响

以矩形表示机床,以有向线段表示机床之间的刀具申请(称作刀具申请线,包括单一线,复选线和争用线),在各矩形之间加入系统中所有的刀具申请线,就构成了刀具申请分配图 TAAG^[64]。在考虑刀具寿命之后,若申请机床所需求的某类刀具实例的剩余寿命不满足其申请,但多把同类刀具的剩余寿命之和却能够满足申请机床的需求时,相应 TAAG 中的申请机床节点可以发出一类新的刀具申请线,请求多把同类刀具全部分配给自己。同时 TAAG 中原有的刀具申请线(包括单一线,复选线和争用线)

将受到此类新刀具申请线的影响，从而表现出一些新的特征。区别于 TAAG 中原有的各种刀具申请线，下面首先给出这类新的刀具申请线的定义。

定义 1 TAAG 中，对于某机床节点发出的多终点的刀具申请线，若只有当所有终点指向的机床中的相关刀具全部被分派给此申请机床时，该机床的刀具申请需求才能够得到满足，则称此类刀具申请线为全选线。

在 TAAG 基础上加入全选线就形成了扩展的刀具申请分配图 E-TAAG，下面给出其形式化定义。

定义 2 $E\text{-TAAG} = (V_M, E_S, E_M, E_A)$,

$V_M = \{M_1, M_2, \dots, M_n\}$ 为机床顶点集合；

$E_S = \{M_i \rightarrow (s)M_j \mid i, j = 1, 2, \dots, k; i \neq j\}$ 为单一线集合，其意义为系统中某机床发出刀具申请，而被申请刀具有且只有 1 把并且被其它机床占用。

$E_M = \{M_i \rightarrow (m)[M_1, M_2, \dots, M_n] \mid i = 1, 2, \dots; i \neq n\}$ 为新复选线集合，其中： $M_i \rightarrow (m_\lambda)[M_k]$ 表示新复选线的第 λ 个全选线分支， $M_i \rightarrow (m_\lambda)M_k$ 表示新复选线的第 λ 个单一线分支。新复选线表征系统中某机床所申请的刀具不止 1 把，且这些刀具实例都被其它机床占用。

$E_A = \{M_i \rightarrow (a)[M_1, M_2, \dots, M_n] \mid i = 1, 2, \dots; i \neq n\}$ 为全选线集合，其中： $M_i \rightarrow (a_\lambda)[M_n]$ 表示全选线的第 λ 个分支。全选线表征对于申请机床节点发出的刀具申请线，只有其所有终点处机床中相关的刀具全部被分派给起点机床时，申请机床的刀具需求才能够得到满足。

定义 3 若 E-TAAG 中存在申请同一类刀具的单一线集合 E_S 或申请同一类刀具的新复选线集合 E_M ，则称集合 E_S 或 E_M 为争用线，表示为： $E_z = \{[M_1, M_2, \dots, M_p] \rightarrow (z)[M'_1, M'_2, \dots, M'_p]\}$ ，即争用线 E_z 的起点集为 $[M_1, M_2, \dots, M_p]$ ，终点集为 $[M'_1, M'_2, \dots, M'_p]$ 。争用线表征为多台机床同时申请同一把刀具或多台机床同时申请多把同类刀具，而被申请的刀具都被其它机床占用。

扩展的刀具申请分配图 (E-TAAG) 示例如图 3.1 所示。

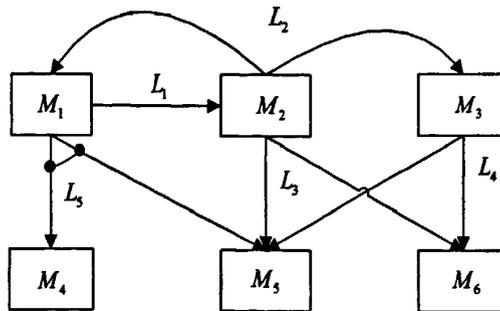


图 3.1 扩展的刀具申请分配图

Fig.3.1 Extended tool applying allocation graph

图 3.1 中 L_1 为单一线, L_2 、 L_3 、 L_4 为新复选线, 且这些刀具申请线终点所指向的机床中相关刀具的寿命都是满足起始点机床申请需要的, 其中 L_3 、 L_4 组成争用线。而刀具申请线 L_5 则是一条全选线, 也即只有 M_4 和 M_5 中被申请的相关刀具全部被分配给 M_1 时才能满足其刀具申请需求 (假设 M_4 和 M_5 中被申请的相关刀具的剩余寿命之和能够满足需要)。

考虑刀具寿命之后, 原有的 TAAG 中的各种申请线将发生相应的变化, 并最终体现在 E-TAAG 中。以下按照申请线种类进行逐一分析。

对于单一线, 若申请线终点指向的机床中相关刀具的剩余寿命满足起始点机床的申请需求时, 该单一线保持不变; 否则, 此单一线将不存在。

对于复选线, 若申请线所有终点指向的机床中相关刀具的剩余寿命均满足起始点机床的申请需求时, 该复选线保持不变; 否则, 复选线可能出现以下 3 种转化情况:

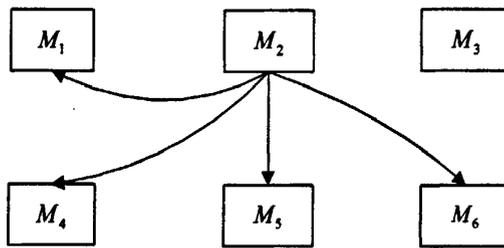


图 3.2 原复选线

Fig.3.2 Original MSL

1) 原复选线转化为单一线

原复选线如图 3.2 所示, M_2 发出一条复选线向相关机床申请某类刀具。当只有 M_4 中被申请刀具的剩余寿命满足需求, 而 M_1 、 M_5 、 M_6 中的被申请刀具剩余寿命不足且三者之和都不能满足申请机床的需求时, 将转化为如图 3.3 所示的单一线。

2) 原复选线转化为新复选线

若 M_4 、 M_5 、 M_6 中被申请刀具的剩余寿命都能够满足 M_2 的申请, 只有 M_1 中的相关刀具剩余寿命不足, 则删掉此复选线分支, 原复选线将转化为如图 3.4 所示的新的复选线。图中的“三角号”表示因为刀具剩余寿命不足而删除此分支。

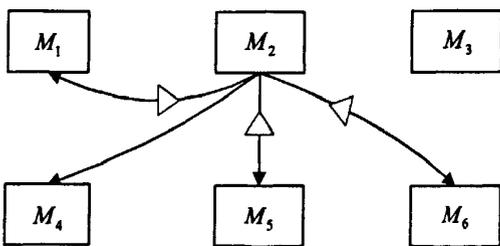


图 3.3 转变为单一线

Fig.3.3 converted to SL

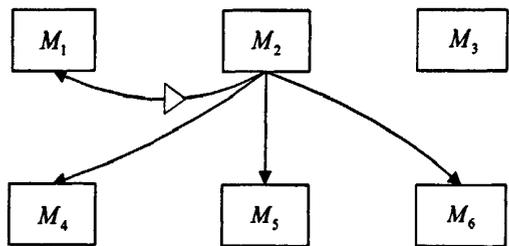


图 3.4 转变为新复选线

Fig.3.4 converted to new MSL

在复选线的分支中，若某些分支的终点所指向的机床中相关刀具的剩余寿命能够满足起始点机床的申请，而其它分支中被申请刀具不满足需求，但是组合起来却能提供给申请机床时，则原复选线将转化为新复选线。如图 3.5 所示， M_6 中的被申请刀具的剩余寿命满足 M_2 的需求，而 M_1, M_4, M_5 中的刀具剩余寿命都不能单独满足 M_2 的申请，但是 M_1 与 M_4 或者 M_4 与 M_5 组合却可以保证 M_2 的需求。

3) 原复选线转化为全选线

若 M_1, M_4, M_5, M_6 中被申请刀具的剩余寿命都不能单独提供给申请机床，当且仅机床中相关刀具全部被分派给 M_2 时才能够满足其申请需求时，原复选线将转化为如图 3.6 所示的全选线。

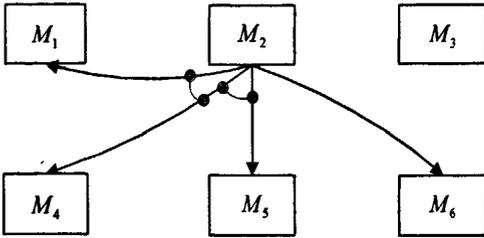


图 3.5 转化为新复选线

Fig.3.5 converted to new MSL

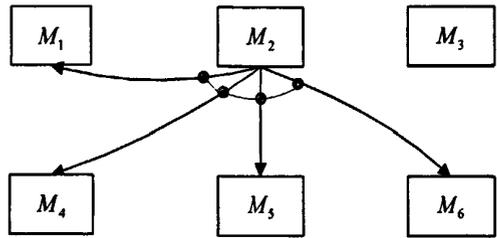


图 3.6 转化为全选线

Fig.3.6 converted to WSL

对于争用线，在考虑刀具寿命时，可以类似地得到转化后的 E-TAAG。

1) 原争用线转化为新的争用线

原争用线如图 3.7 所示，当 M_3 中的被申请刀具的剩余寿命不满足 M_4 和 M_5 的申请时，则删除相关的刀具申请线，也即原争用线将转化为新的争用线，如图 3.8 所示。

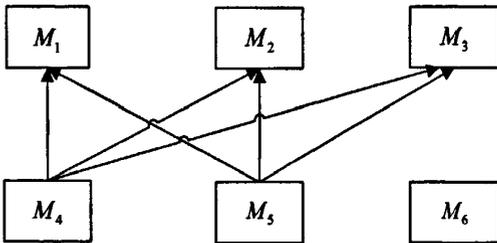


图 3.7 原争用线

Fig.3.7 converted to CL

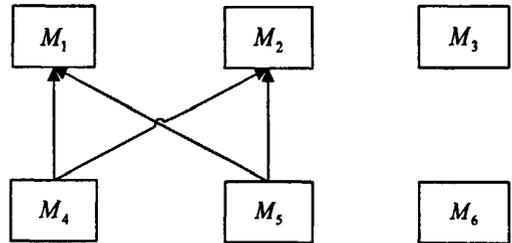


图 3.8 转化为新的争用线

Fig.3.8 converted to new CL

2) 原争用线转化为“争用+新复选”线

当 M_2 和 M_3 中的被申请刀具的剩余寿命均不满足申请机床的需求（假设二者同时分派给所需机床能够满足），而 M_1 中的相关刀具寿命能够满足时，原争用线将转化为“争用+新复选”线，如图 3.9 所示。

3) 原争用线转化为“争用+全选”线

当 M_1, M_2, M_3 中的被申请刀具寿命均不满足申请机床的需求，但三者同时分

派给所需机床能够满足时，原争用线将转化为“争用+全选”线，如图 3.10 所示。

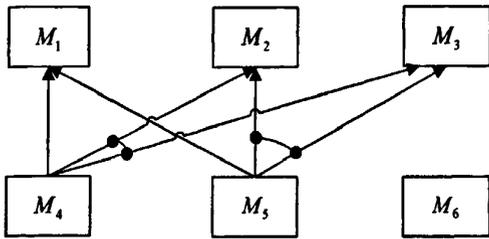


图 3.9 转化为“争用+新复选”线

Fig.3.9 converted to CL& new CL

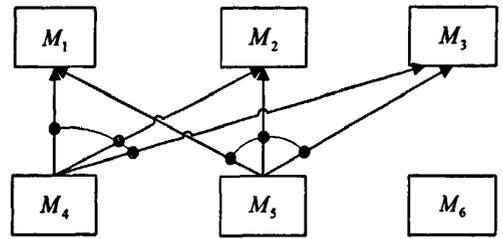


图 3.10 转化为“争用+全选”线

Fig.3.10 converted to CL&WSL

根据上述分析可以看出，引入全选线之后所形成的扩展的刀具申请分配图 (E-TAAG) 可以很好地描述 FMS 中考虑刀具寿命因素的刀具资源的申请和分配状况。

3.1.2 E-TAAG 的构建

扩展的刀具申请分配图 E-TAAG 的构建流程如图 3.11 所示。

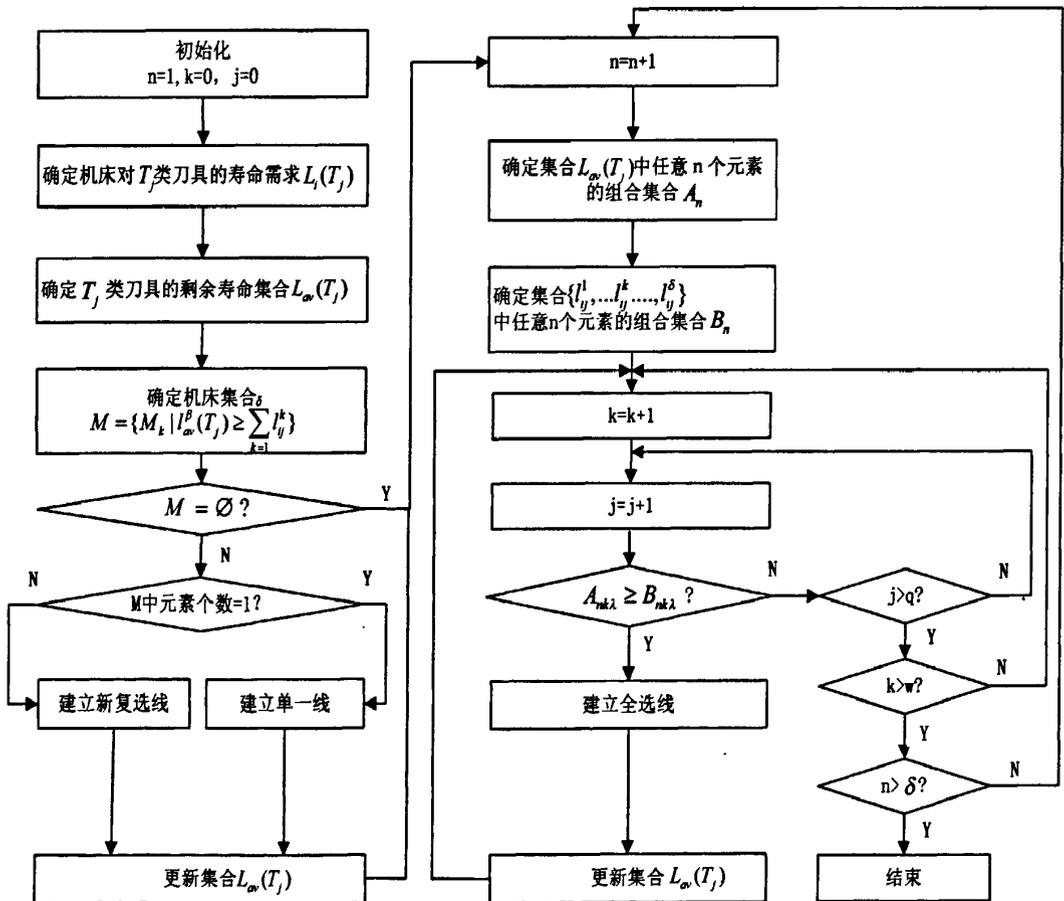


图 3.11 E-TAAG 构建流程图

Fig3.11 Flow chart of E-TAAG construction

Step1: 确定机床 M_i 上工序使用 T_j 类刀具的时间集合 $L_i(T_j) = \{l_{ij}^k \mid k=1,2,\dots,\delta\}$, 其中: l_{ij}^k 为机床 M_i 上工序使用 T_j 类刀具的各分工步的加工时间;

Step2: 确定 T_j 类刀具在其他机床上的剩余寿命集合 $L_{av}(T_j) = \{l_{av}^\beta(T_j) \mid \beta=1,2,\dots,m\}$, 其中 $l_{av}^\beta(T_j)$ 为机床 M_β 上 T_j 类刀具的剩余寿命;

Step3: 若存在一个 $l_{av}^\beta(T_j) \in L_{av}(T_j)$ 满足: $l_{av}^\beta(T_j) \geq \sum_{k=1}^{\delta} l_{ij}^k, \beta=1,2,\dots,m$, 则建立单一选线 $M_i \rightarrow (s)M_\beta$, 更新集合 $L_{av}(T_j)$ 即去掉满足条件的 $l_{av}^\beta(T_j)$, 否则 *step4*;

Step4: 若存在多个 $l_{av}^\beta(T_j) \in L_{av}(T_j)$ 满足: $l_{av}^\beta(T_j) \geq \sum_{k=1}^{\delta} l_{ij}^k, \beta=1,2,\dots,m$, 则建立新复选线 $M_i \rightarrow (m)[M_1, M_2, \dots, M_\beta, \dots], \beta \in (1,2,\dots,m), \beta \neq i$, 更新集合 $L_{av}(T_j)$, 即去掉满足条件的 $l_{av}^\beta(T_j)$, 否则 *step5*;

Step5: 确定 $L_{av}(T_j)$ 中 n 个元素的组合集合 $A_n = \{A_{n1}, \dots, A_{nk}, \dots, A_{nw}\}$, $n=2,3,\dots,\delta$, $k=1,2,\dots,w$ 且 $A_{nk} = \{A_{nk1}, \dots, A_{nk\lambda}, \dots, A_{nkn}\}$ 。其中: $A_{nk\lambda} \in L_{av}(T_j) (\lambda=1,2,\dots,n)$ 同时 A_{nk} 满足以下两个条件:

$$1) \sum_{\lambda=1}^n A_{nk\lambda} \geq \sum_{k=1}^{\delta} l_{ij}^k,$$

$$2) \min\{A_{nk1}, \dots, A_{nk\lambda}, \dots, A_{nkn}\} \geq \min\{l_{ij}^1, l_{ij}^2, \dots, l_{ij}^{\delta}\}$$

Step6: 确定集合 $\{l_{ij}^1, l_{ij}^2, \dots, l_{ij}^{\delta}\}$ 中任意 n 个元素的组合集合 $B_n = \{B_{n1}, \dots, B_{nj}, \dots, B_{nq}\}$, $n=2,3,\dots,\delta$, $j=1,2,\dots,q$ 且 $B_{nj} = \{B_{nj1}, \dots, B_{nj\lambda}, \dots, B_{njn}\}$ 。其中: $B_{nj\lambda} = \sum_{k \in \varphi} l_{ij}^k$, $\varphi=1,2,\dots,\delta$;

Step7: 若 $\exists A_{nk}$ 且有 $\forall A_{nk\lambda} \in A_{nk}$ 满足: $A_{nk\lambda} \geq B_{nk\lambda}$, 则建立全选线 $M_i \rightarrow (a)[M_1, M_2, \dots, M_\beta, \dots], \beta \in (1,2,\dots,m), \beta \neq i$ 。

3.2 刀具流死锁性质分析

在建立了刀具流的 E-TAAG 后, 结合图论的相关概念和理论, 即可对刀具流死锁进行深入剖析。

定理 1 刀具流发生死锁的充分必要条件是: 对应的 E-TAAG 中存在闭合的强连通图 K , K 中任意机床节点均有刀具申请线, 且至少有一个刀具申请线落在 K 中。其中, 刀具申请线落于 K 中是指: 对于全选线, 要求其至少有一个分支的终点落于 K 中; 对于新复选线要求其全选线分支至少有一个终点落在 K 中, 而对单一选线分支要求其终点均落在 K 中; 对于单一选线要求其终点均落于 K 中。

证明:

充分性: 若 E-TAAG 中存在一个闭合的强连通图 K , 则 K 中每个机床的可达节点集恰好是 K 中机床节点集, 又单一选线、复选线, 其所有终点均落于 K 中, 全选线, 其至少有一个分支的终点落于 K 中, 这暗含 K 中每台占有刀具的机床都在等待 K 中

其它机床释放刀具，而K中每台占用刀具的机床都至少被一个其它机床等待（否则一个机床将不在它自己的可达节点集中），因而也就有一组机床集合，集合中每台机床都在等待集合中其它机床释放刀具，据死锁定义，刀具流发生死锁。

必要性：假设刀具流死锁，则必定存在一组机床集合，集合中的每台机床都在等待集合中其他机床释放刀具。在E-TAAG中就表现为集合中每台机床节点都至少有一个出线申请指向集合中的其他机床节点，同时也暗含E-TAAG中一定存在一个回路。现构建 $E-TAAG_{DL} = (V_{MDL}, E_{SDL}, E_{MDL}, E_{ADL}, E_{ZDL})$ ，且 $V_{MDL} \subset V_M$ 为死锁的机床节点集合， $E_{SDL} \subset E_S$ 为死锁机床间的单一线集合， $E_{ADL} \subset E_A$ 为死锁机床间的全选线集合， $E_{MDL} \subset E_M$ 为死锁机床间的新复选线集合。应当指出的是：若存在单一线 $M_k^i \rightarrow (s)M_j \in E_S$ 且 $M_k \in V_{MDL}$ ，则必有 $M_j \in V_{MDL}$ ，也即对于单一线要求其终点均落于K中；若存在全选线 $M_k \rightarrow (a)[M_1, M_2, \dots, M_n] \in E_A$ ，则至少有 $M_k \rightarrow (a_\lambda)[M_\tau] \in E_{ADL}, \tau = 1, 2, \dots, n$ ，也即对于全选线，要求其至少有一个分支的终点落于K中；若存在新复选线 $M_k \rightarrow (m)[M_1, M_2, \dots, M_n] \in E_M$ ，则至少有 $M_i \rightarrow (m\alpha'_\lambda)[M_k] \in E_{MDL}$ ， γ 为第 λ 条全选线的分支号，即复选线的第 λ 条全选线要求至少有一个终点落在K中，同时又有 $M_i \rightarrow (m\alpha_\lambda)[M_k] \in E_{MDL}, \lambda = 1, 2, \dots$ ，即所有单一线分支要求终点均落在K中。

现设 $C = \{C_1, C_2, \dots, C_q\}$ 是 $E-TAAG_{DL}$ 中的强连通图集合，其中 C_i 为一个强连通图且包含一个最大的可达节点子集，其中 C_i 与 C_j 之间没有可达的节点。定义 $G = \{C, \partial_c\}$ ，如图3.12所示。

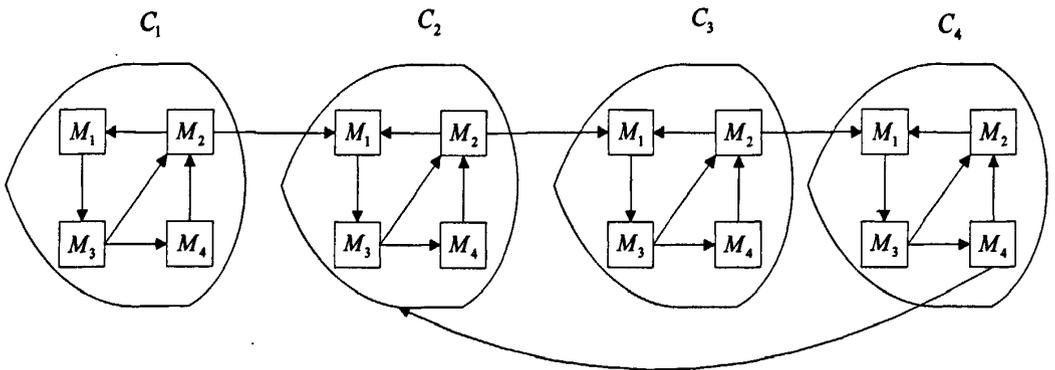


图 3.12 定理 1 的矛盾证明图

Fig.3.12 Contradiction proof graph of theorem 1

$(C_i, C_j) \in \partial_c$ 当且仅当对 $(M_u, M_v) \in E_{LDL}$ 有 $M_u \in C_i, M_v \in C_j, i \neq j$ ，也即机床 M_u 等待机床 M_v 释放刀具，而 M_u 与 M_v 又不相互可达。若不存在一个闭合的强连通图，则每个强连通图 C_i 都至少有一个出线申请，这与假设 C_i 有一个最大的可达节点集矛盾，因而闭合强连通图一定存在。

上述定理将考虑刀具寿命后的刀具流的死锁现象与 E-TAAG 紧密联系起来, 为后续的死锁检测、无死锁调度等研究奠定了理论和技术基础。为便于叙述, 以下将定理 1 中对应的 E-TAAG 称为死锁图 (Deadlock-TAAG, 简称 D-TAAG)。反之, 若 E-TAAG 不满足定理 1 的死锁条件, 则将 E-TAAG 称为非死锁图, 简称为 \bar{D} -TAAG。

推论 1 若 E-TAAG 是死锁图, 则 E-TAAG 中一定存在回路。

若 E-TAAG 是死锁图, 也即刀具流发生死锁, 则 E-TAAG 中一定存在强连通图 K, K 中任意两个节点都是相互可达的, 故必存在回路 BC 经过 K 中所有各点, 否则必有一回路不包含某一节点 v, 并且 v 与回路 BC 上的各节点就不是相互可达的, 与强连通图条件矛盾。

推论 2 如果系统中任意机床均有刀具申请, 则该系统刀具流必定死锁。

对此情况建立相应的 E-TAAG, 则该 E-TAAG 即为一个闭合的强连通图, 且该闭合的强连通图中任意机床节点均有刀具申请线, 并至少有一个刀具申请线落在 E-TAAG 中, 根据定理 1 可知该系统刀具流发生死锁。

推论 3 D-TAAG 中加入任何新的单一线、全选线及新复选线后形成的新的 E-TAAG 仍然是 D-TAAG; \bar{D} -TAAG 删去任何原有的单一线、全选线及新复选线后形成的新的 E-TAAG 仍然是 \bar{D} -TAAG。

此推论说明只有改变原有强连通图也即死锁节点集中的刀具申请线才可能消除死锁, 同时也指出刀具流的死锁状态不会因为加入新的刀具申请而消除; 另外它也说明了对 \bar{D} -TAAG 的删除刀具申请线操作不影响该图的非死锁性。

定理 2 若 E-TAAG 为 \bar{D} -TAAG, 且图中存在无刀具申请线的机床节点集 $M = [M_1, M_2, \dots, M_p]$, 那么删去指向 M 的单一线、全选线及新复选线后形成的新的 E-TAAG 仍为 \bar{D} -TAAG。

证明: 假设 E-TAAG 为 \bar{D} -TAAG, 则由定理 1 的必要性证明可知, E-TAAG 中一定存在闭合的强连通图也即死锁节点集 DM, 为了验证删除申请线对 E-TAAG 死锁性的影响, 申请线的起点必须位于死锁节点集 DM 内。

1) 对单一线, 应为图 3.13 所示情形。

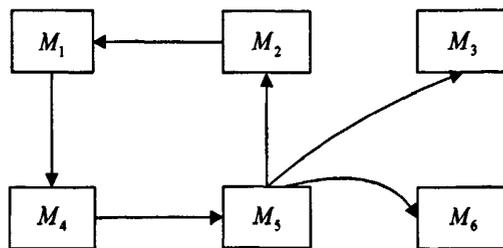


图 3.13 M 内的单一线

Fig.3.13 SL in M

$DM = \{M_1, M_2, M_4, M_5\}$ 为死锁节点集, $M = \{M_3, M_6\}$ 为无出线节点集, 且指向 M 的单一线 $M_5 \rightarrow (s)M_3$ 和 $M_5 \rightarrow (s)M_6$ 的起点机床为 $M_5 \in DM$, 删除此两条单一线, 死锁节点集 DM 不发生变化, 故 E-TAAG 仍为死锁图。

2) 对全选线, 如图 3.14 所示, 存在全选线 $M_5 \rightarrow (a)[M_2, M_3, M_6]$, 起点集位于 DM 内, 且其分支 $M_5 \rightarrow (a_1)[M_3]$ 和 $M_5 \rightarrow (a_2)[M_6]$ 指向无出线节点集 M , 删除此两分支后, 全选线 $M_5 \rightarrow (a)[M_2, M_3, M_6]$ 仍有一分支 $M_5 \rightarrow (a_3)[M_2]$ 的终点落在 DM 内, 故 E-TAAG 的死锁性不变。

3) 对新复选线, 如图 3.15 所示。存在新复选线 $M_5 \rightarrow (m)[M_1, M_2, M_3, M_6]$, 起点集位于 DM 内, 且 $M_5 \rightarrow (ma_1)[M_2, M_3, M_6]$ 为其全选线分支, $M_5 \rightarrow (ms_1)M_1$ 为其单一线分支, 删除指向无出线节点集 M 的分支 $M_5 \rightarrow (ma_1)[M_3]$ 和 $M_5 \rightarrow (ma_2)[M_6]$ 后, 复选线的全选线分支仍有一个终点落在 DM 内, 单一线分支不发生改变, 故 E-TAAG 仍为死锁图。

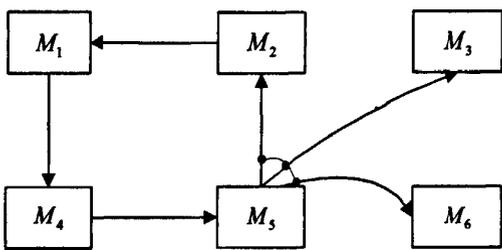


图 3.14 M 内的全选线

Fig.3.14 WSL in M

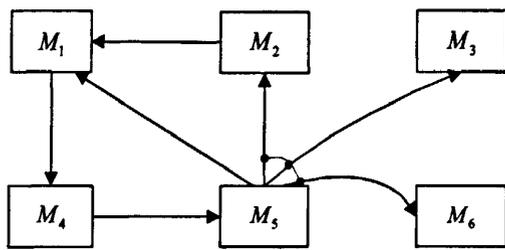


图 3.15 M 内的新复选线

Fig.3.15 new ML in M

4) 若删除的单一线 $M_5 \rightarrow (s)M_3$ 或 $M_5 \rightarrow (s)M_6$ 为争用线的组成部分, 删除的全选线的两个分支 $M_5 \rightarrow (a_1)[M_3]$ 和 $M_5 \rightarrow (a_2)[M_6]$ 为争用线的组成部分, 删除的新复选线的全选线分支 $M_5 \rightarrow (ma_1)[M_3]$ 和 $M_5 \rightarrow (ma_2)[M_6]$ 为争用线的组成部分, 即存在争用线 $[M'_1, M'_2, \dots, M'_p] \rightarrow (z)[M_3]$ 或 $[M'_1, M'_2, \dots, M'_p] \rightarrow (z)[M_6]$, 那么无论争用线的起点集是否位于 DM 内, 则删除指向 M 的单一线, 全选线或新复选线后, 争用线改变为 $[M'_1, M'_2, \dots, M'_p] \rightarrow (z)[M_3]$, 相当于在死锁节点集 DM 内加入了新的申请线, 根据定理 1 的推论 3 可知, E-TAAG 仍为死锁图。

综上, 删去指向 M 的单一线、全选线及新复选线后所形成的新的 E-TAAG 与原 E-TAAG 具有相同的死锁性。

定理 3 设某 E-TAAG 是 \bar{D} -TAAG, $M = [M_1, M_2, \dots, M_p]$ 为图中的无刀具申请线的机床节点集, $\bar{M} = [M'_1, M'_2, \dots, M'_p]$ 为图中的有刀具申请线的机床节点集, 则有:

1) $M \neq \phi$;

2) $\exists M_u \in \bar{M}$ 满足: $D'_u \subseteq M, \forall i=1, 2, \dots, r, \exists D'_u(\lambda) \subseteq M, \forall j=1, 2, \dots, q$ 。其中 D'_u 为 M_u 发出的第 i 条申请线的终点节点集, r 为 M_u 发出的单一线和全选线的总数; $D'_u(\lambda)$ 为

M_u 发出的第 j 条申请线某一个分支的终点节点集, q 为 M_u 发出的新复选线总数, λ 为 M_u 发出的第 j 条申请线的分支号。

3) 满足条件的 M_u 获得 M 中节点所代表的机床释放的刀具后重新生成的 E-TAAG 仍然是 \bar{D} -TAAG。

证明:

1) 假设 $M = \phi$, 则由定理 1 的推论 1 可知, E-TAAG 中所有机床节点均有刀具申请线, 因此该 E-TAAG 是 \bar{D} -TAAG, 与假设矛盾, 故 $M \neq \phi$ 。

2) 假设满足条件的 M_u 不存在, 即 $\forall M_u \in \bar{M}$ 都有: 对其发出的单一线有 $D'_u \subseteq M$ 不成立, 则 $D'_u \cap M = \phi$, 说明单一线终点均落在 \bar{M} 中; 对全选线有 $D'_u \subseteq M$ 不成立, 说明全选线至少有一个分支的终点落在 \bar{M} 中; 对新复选线 $\exists D'_u(\lambda) \subseteq M$ 不成立, 说明其全选线分支至少有一个终点落在 \bar{M} 中, 而单一线分支全部落在 \bar{M} 中。根据定理 1, 该 E-TAAG 为死锁图, 与题设矛盾, 所以一定存在满足条件的 M_u 。

3) 设 M_u 满足 2), 即有 $D'_u \subseteq M, \forall i=1,2,\dots,r, \exists D'_u(\lambda) \subseteq M, \forall j=1,2,\dots,q$ 。当 M_u 获得 M 中节点所代表的机床释放的刀具后必须根据分派后刀具资源的分布状况重新生成系统的 E-TAAG, 由于 M_u 满足 2), 所以 M_u 申请的刀具都能够得到满足, 故在新的 E-TAAG 中 M_u 为无出线节点机床。事实上就是从原 E-TAAG 中删去了指向无出线节点集 M 的单一线 $M_u \rightarrow (s)M_k, k=1,2,\dots,p$, 新复选线 $M_u \rightarrow (m)[M_1, M_2, \dots, M_p, \dots]$, 全选线 $M_u \rightarrow (a)[M_1, M_2, \dots, M_p]$, 由定理 1 的推论 3 可知, 原 E-TAAG 为 \bar{D} -TAAG 时, 从中删去任何原有的单一线、全选线及新复选线后形成的新的 E-TAAG (设为 A) 是 \bar{D} -TAAG。如果删除的申请线是争用线 $[M_u, M'_1, M'_2, \dots, M'_p] \rightarrow (z)[M_1, M_2, \dots, M_p, \dots]$ 的组成部分, 还需考虑争用线改变后的 E-TAAG (设为 B) 是否仍为非死锁图。

将争用线 $[M_u, M'_1, M'_2, \dots, M'_p] \rightarrow (z)[M_1, M_2, \dots, M_p, \dots]$ 等效为以下一组新复选线:

$$M_u \rightarrow (m)[M_1, M_2, \dots, M_p, \dots]$$

$$M'_1 \rightarrow (m)[M_1, M_2, \dots, M_p, \dots]$$

$$M'_2 \rightarrow (m)[M_1, M_2, \dots, M_p, \dots]$$

$$\vdots$$

$$M'_p \rightarrow (m)[M_1, M_2, \dots, M_p, \dots]$$

由于 M_u 获得了争用刀具, 故争用线发生改变, 即 $M'_k (0 < k < p)$ 发出的上述复选线变化如下:

$$M'_1 \rightarrow (m)[M_u, \dots]$$

$$M'_2 \rightarrow (m)[M_u, \dots]$$

$$\vdots$$

$$M'_p \rightarrow (m)[M_u, \dots]$$

不妨设 B 为死锁图, 由死锁图定义可知其中必定存在死锁机床节点集 DM , 且 M_u

一定为 DM 中的死锁机床节点, 也即集合中的每台机床节点都有出线申请, 显然与 M_u 为无出线节点机床矛盾, 故 B 为非死锁图。条件 3) 得证。

3.3 刀具流死锁检测算法

根据定理 1, 建立刀具流死锁检测算法如下:

输入: $E\text{-TAAG} = \{V_M, E_L\}$;

输出: 死锁, 无死锁;

Step1: 确定 $E\text{-TAAG}$ 内的强连通图集合: $C = \{C_1, C_2, \dots, C_q\}$;

Step2: 若 $|C|=1$, 转到 Step 5. 否则若 $|C_i|=1, \forall i=1, 2, \dots, q$, 返回无死锁;

Step3: 构建 $G = \{C, \partial_c\}$: $C = \{C_1, C_2, \dots, C_q\}$ 和

$\partial_c = \{(C_i, C_j) : (M_u, M_v) \in E_L \text{ and } M_u \in C_i, M_v \in C_j, i \neq j\}$;

Step4: For $i=1 \dots q$

if $|C_i| > 1$ and $(C_i, C_j) \notin \partial_c, \forall j=1, 2, \dots, q$, go to Step5

End For;

Step5: 检测落在强连通图 C_i 的出线申请中, 新复选线的单一线分支终点是否全部落在 C_i 内, 新复选线的全选线分支至少有一个终点落在 C_i 内, 若是则返回死锁, 否则返回无死锁。

Step1 确定 $E\text{-TAAG}$ 内的强连通图集合, 其算法能在一些文献[]中获得; Step2 检测强连通图, 若只有一个强连通图 C_i , 则其一定是闭合的强连通图, 那么转到 Step5, 如果强连通图的个数大于 1 个, 且每个强连通图内只有一个节点, 那么没有节点相互可达, 也就没有死锁; Step3 构建了一个图来定义强连通图之间的关系; Step4 寻找至少有两个节点且无出线申请的强连通图, 也即用来判断是否存在闭合的强连通图, 若不存在则无死锁, 若存在则转到 Step5 去判断落在强连通图 C_i 的出线申请中, 新复选线的终点是否全部落在 C_i 内, 若是, 则返回死锁, 否则返回无死锁。

刀具流死锁检测算法的建立为后续刀具流无死锁调度算法的顺利实施准备了技术支撑。

3.4 刀具流无死锁调度算法的实现

3.4.1 死锁避免策略的基本思想

定理 1 表明各机床欲申请刀具的占用和需求之间的矛盾是刀具流发生死锁的主要原因。D-TAAG 是实际运行中的刀具流死锁状态的图论域体现, 该情况的出现一般是在机床选择待加工工序时刻。

在初始 $E\text{-TAAG}$ 为 $\bar{D}\text{-TAAG}$ 的前提下, 如果采用了不合理的刀具分派方案, 后续的 $E\text{-TAAG}$ 将仍然可能形成 D-TAAG, 显然, 刀具分派的不合理是刀具流死锁的另一

重要根源。为此，为使各个刀具申请都能顺利地满足并且不再导致后续 E-TAAG 形成死锁，就必须按照一定的规则合理分派刀具，合理制定刀具分派策略。

根据上述分析，从刀具流死锁的根源出发，本文给出一类两级死锁避免策略如图 3.16 所示。

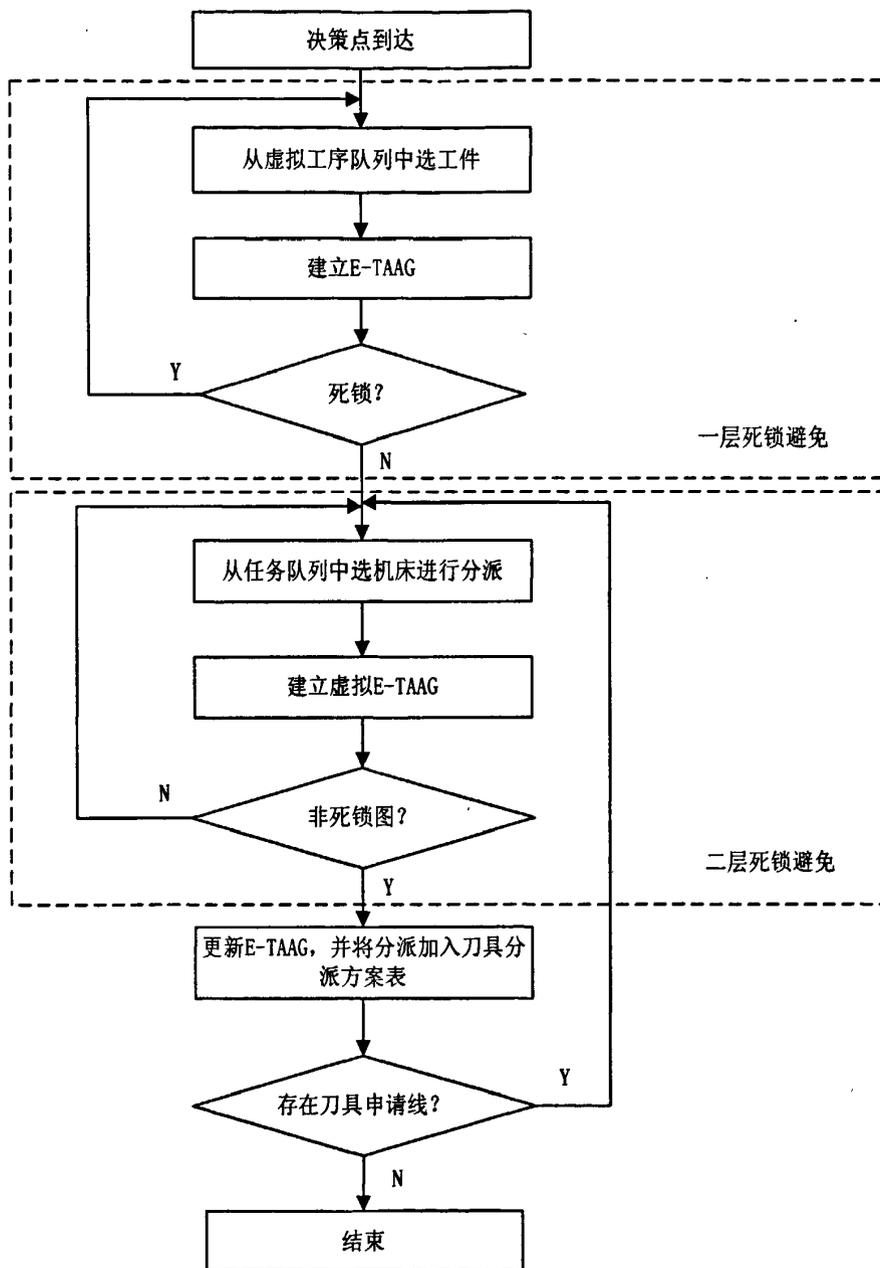


图 3.16 两层死锁避免策略原理图

Fig.3.16 Schematic diagram of double-layer deadlock avoidance strategy

在决策点^[135]处，首先判断哪些机床当前能够进行下一工序的决策，然后从相应的虚拟工序队列中进行选择。根据所选加工工序的工艺信息及当前的刀具分派状态即

可确定所选工序对刀具资源的需求状态,从而建立扩展的刀具申请分配图 E-TAAG。然后根据死锁判定定理即可进行死锁判定,根据判定结果确保所选工序不会导致死锁,此即为第一层死锁避免。它能够保证各机床的下道工序的加工过程一定存在着可行的刀具分派方案。由于同一个 E-TAAG 一般可以对应着多个刀具分派方案,其中一些是不可行方案,所形成的刀具分派一定会导致后续死锁。因此,还要进行第2层次的死锁控制,即进行无死锁的刀具分派方案的求解。也即每类刀具从其任务队列中选择要分派的机床,它应该保证此次刀具分派形成的新 E-TAAG 是 \bar{D} -TAAG,从而保证各机床下道工序加工的顺利进行,将刀具分派任务加入刀具分派方案表,直到每类刀具的任务队列都为空,此时刀具分派方案表中存储的可行的刀具分派进程即是无死锁刀具分派方案。

3.4.2 无死锁调度算法

基于两层死锁避免策略,可以建立刀具流无死锁调度算法如图 3.17 所示。

Step1: 系统初始化,并令机床上工序的开始加工时刻 t 和结束时刻 t_1 为 0,机床输入缓冲区中工序的加工结束时刻 t_2 为无穷;

Step2: 机床开始加工,并令 t =系统当前时刻, $t_1=t$ +当前工序加工时间;

Step3: 当 t_1 到达时,若机床输入缓冲区有待加工工件,则转 *step4*,否则等待系统决策点 t_d 到达时为其决策下一待加工工件;

Step4: 若待加工工序所需刀具已经备好,则 *step5*,否则等待刀具的到达,并计算等待刀具的时间 Δt ;

Step5: 输入缓冲区中的工件搬上机床并且开始加工,并令 $t=t_1+\Delta t$, $t_2=t$ +待加工工序加工时间,同时系统的决策点 t_d 到达;

Step6: 若可决策机床其虚拟工序队列为空,则结束,否则 *step8*;

Step7: 按机床选工件规则从其虚拟工序队列中选择工序,并建立相应的 E-TAAG 进行死锁判定,若死锁,则机床重新选择工序,否则重构虚拟工序队列,并转 *step9*;

Step8: 按无死锁要求分派刀具;

1) 刀具分派方案表 TAP 置空;

2) 生成无刀具申请线的机床节点集 $M=[M_1, M_2, \dots, M_p]$ 和有刀具申请线的机床节点集 $\bar{M}=[M'_1, M'_2, \dots, M'_p]$, 若 $\bar{M}=\phi$, 则转到 8); 若 $\bar{M} \neq \phi$, 则寻找 M_{ui} ;

3) 若 M_{ui} 的数量为 1, 则 5); 否则, 确定 M 中各节点前的机床队列 $M_{u1}, M_{u2}, \dots, M_{ui}$;

4) 按刀具选机床规则确定刀具优先分派的机床 M_{ui} , 若此时机床 M_{ui} 所对应的各刀具恰好是其所需求的, 则将刀具分派方案加入刀具分派方案表, 否则 5);

5) 按“非争用刀具节点优先规则+刀具剩余寿命最小规则”分派刀具给 M_{ui} , 同时将刀具分派方案加入 TAP;

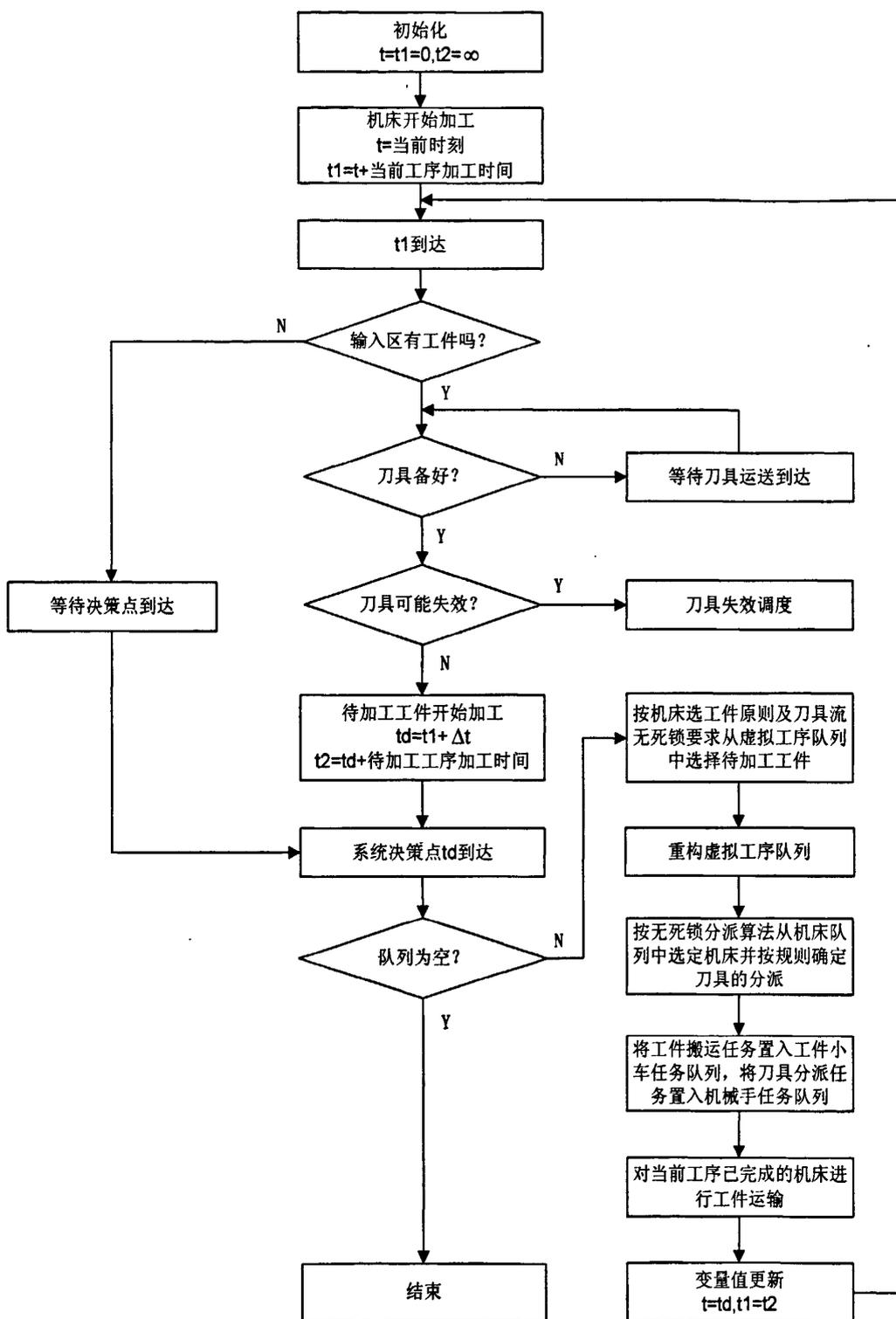


图 3.17 工序备刀下刀具流无死锁调度算法流程图

Fig.3.17 Flow chart of deadlock-free scheduling in case of procedure provisioning tools

6) 根据刀具分派后的系统刀具分布, 重新生成 E-TAAG;

7) 重复 1) —6);

8) 输出刀具分派方案表 TAP, 结束。

Step9: 将工件搬运任务置入工件小车的任务队列, 将 TAP 表中的刀具分派任务置入运刀机械手任务队列;

Step10: 对当前已经加工完成的工序进行工件运输;

Step11: 更新变量值, 令 $t1=t2$ 。

3.4.3 刀具失效调度

刀具失效主要指机床加工过程中的刀具磨损和各种偶然因素引起的刀具破损。在柔性制造系统中, 当刀具失效发生时, 不仅降低了机床利用率, 而且还会影响到全局调度过程, 因此, 必须进行刀具失效调度, 其调度的原则是个别刀具的失效要尽可能不影响或较少地影响系统中其它机床的加工进程。根据此调度原则, 给出工序备刀情况下的刀具失效调度流程如图 3.18 所示。

机床 M_i 正在使用刀具 T_j^1 加工工件时, 若刀具 T_j^1 发生磨损失效, 则首先检查本地机床刀库, 如果有备用刀具则替换刀具 T_j^1 , 若没有则备用刀具一定在中央刀库中 (因为工序备刀情况下, 只有所需要全部刀具准备好后工件才开始加工, 但是由于刀具库容量限制, 可以将所需要的备用刀具暂放在中央刀库中), 此时可向中央刀库请求运刀。

若刀具 T_j^1 发生破损, 此时应根据严重程度进行处理, 对刀具折断情况, 由于这类破损往往对工件加工表面有较大影响, 需对工件进行额外检查或处理, 所以采取的措施是把工件运到工件库的故障缓冲区, 并释放此加工工序所占用的全部非 T 域刀具; 对崩刃、碎断和裂纹等破损情况, 则机床 M_i 应根据现有刀具资源的剩余寿命情况及申请分配状况重新建立扩展的刀具申请分派图 E-TAAG, 若 E-TAAG 为 D-TAAG, 说明不存在可行的刀具分派方案分派备用刀具给机床 M_i , 此时应向系统申请备用刀具; 若 E-TAAG 为 \bar{D} -TAAG, 且发出的为单一线或全选线, 说明有唯一分派方案满足机床 M_i 的申请, 生成刀具机械手附加任务, 等待运输; 如果发出的为新复选线, 说明系统内有多把适用的备用刀具 (可能分布在刀具进出站、中央刀库、其它机床刀库内), 且它们可能正在被使用, 或已经被其它机床预订, 显然, 在满足加工要求前提下, 为了提高系统的响应速度、减少等待刀具的时间, 应该以获得备用刀具的时间最短为选择原则。故需要确定机床 M_i 发出的新复选线落在无出线的机床节点集中的分支 (保证按此分支分派后的 E-TAAG 为 \bar{D} -TAAG), 计算各分支获得备用刀具的时间, 选择时间值最小的新复选线分支, 也即按此分支分派刀具, 机床 M_i 的等刀时间最短, 生成刀具机械手附加任务, 并等待执行, 刀具机械手将备用刀具运至机床 M_i 后, 破损的刀具被更换。从而以个别机床的短时间等待为代价来换取全局的协调。

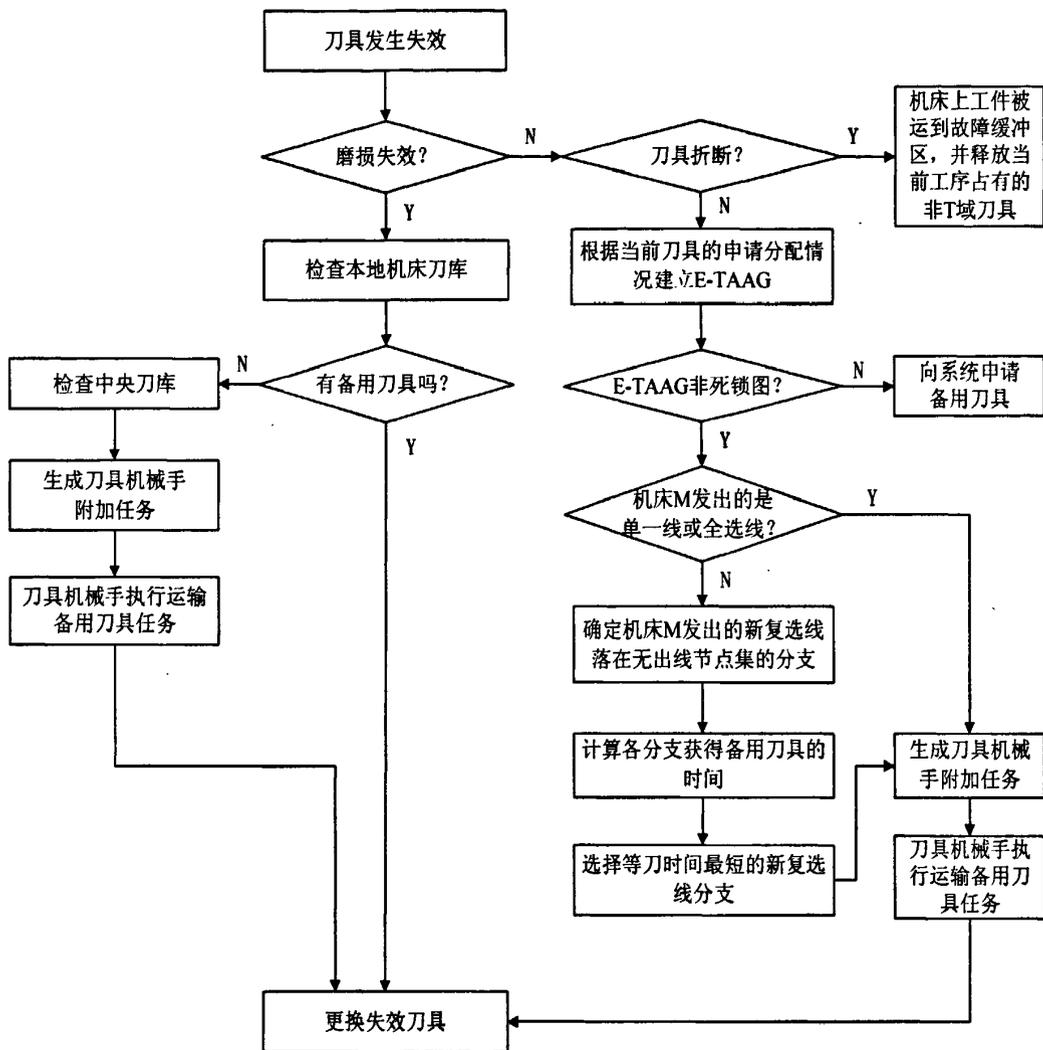


图 3.18 工序备刀下刀具流失失效调度流程图

Fig.3.18 Scheduling flow chart with tools failure in case of procedure provisioning tools

3.4.4 调度规则

现有的一些调度策略和规则只局限于工件流或刀具流内部，没有考虑二者的相互作用和影响，不利于改善和提高系统综合性能。故需要建立着眼于系统整体性能的新的调度规则，也即工件选择规则和刀具分派规则，分别应用于机床选工件和刀具选机床的调度决策，其中机床选工件规则引用文献[64]中的刀具外借量最少规则，而对于刀具选机床，本文给出一种新的调度规则—目标机床当前加工完成时间最早规则 (Earliest Finish Time for Current Operation, EFCO)。新规则的突出特点在于它着眼于系统整体性能的提高，充分考虑了工件流和刀具流的相互作用关系，较之传统的调度规则更为合理，能够有效地提高调度系统的各项性能指标。

EFCO 规则的基本思想是：刀具优先分派给机床队列（任务）中当前工序加工完成时间最早的机床 M_i 。该策略在决策刀具的分派时考虑了对工件流效率的影响，使得系统中最迫切需要刀具的机床能迅速得到刀具，从而其待加工工序能尽早进入加工状态，减少了机床等待刀具的时间，提高了系统的生产效率。其基本表达式如下：

$$\min(t(i)), M_i \in VM$$

其中：VM 为刀具 T_j 的机床队列集合， $t(i)$ 为 M_i 的当前工序的完工时间。

3.5 实例分析

为验证刀具流死锁避免策略的有效性，以下给出了一个刀具分派示例，图 3.19 是给定的一个非死锁的 E-TAAG，图中的申请线情况如下：

单一线 $M_1 \rightarrow (s)M_4$ ，申请刀具 T_2 ；

单一线 $M_2 \rightarrow (s)M_6$ ，申请刀具 T_4 ；

全选线 $M_4 \rightarrow (a)[M_2, M_5]$ ，申请刀具 T_3 ；

新复选线 $M_1 \rightarrow (m)[M_2, M_3]$ ，申请刀具 T_1 ；

新复选线 $M_5 \rightarrow (m)[M_2, M_3, M_6]$ ，其全选线分支 $M_5 \rightarrow (ma_1)[M_3, M_6]$ ，单一线分支 $M_5 \rightarrow (ms_1)[M_2]$ ，申请刀具 T_1 ；

争用线 $[M_1, M_5] \rightarrow (z)[M_2, M_3]$ 由复选线 $M_1 \rightarrow (m)[M_2, M_3]$ 和复选线的全选线分支 $M_5 \rightarrow (ma_1)[M_3]$ 组成，争用刀具 T_1 。

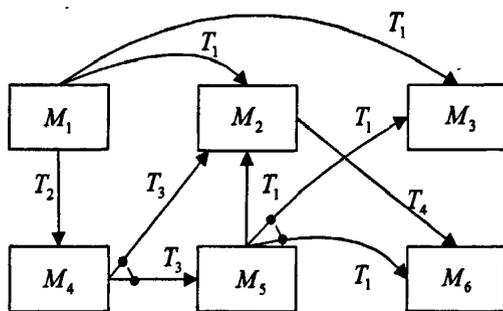


图 3.19 非死锁的 E-TAAG

Fig.3.19 E-TAAG of deadlock-free

针对图 3.19 所示的刀具申请分配状况，具体的刀具无死锁分派过程如下：

1) 初始化刀具分派方案表 TAP；

2) 根据图 3.19 所示的扩展刀具申请分配图 E-TAAG 可知， $M = [M_3, M_6]$ 为无刀具申请线节点集，而 M_2 和 M_5 获得 M 上的刀具 T_4, T_1 后都可以进行下道工序的加工，按规则选择一种进行分派，设为 M_2 获得 T_4 ，于是在 TAP 中加入一条分派顺序： $M_6[T_4] \rightarrow M_2$ ，同时删去单一线 $M_2 \rightarrow (s)M_6$ ；

3) 重新生成 E-TAAG 如图 3.20(a)所示， $M = [M_2, M_3, M_6]$ ，而 M_5 获得 M_2 上的刀

具 T_1 或者同时获得 $[M_3, M_6]$ 上的刀具 T_1 后都可以进行下道工序的加工, 任选一种分派方式, 令 M_2 上的刀具 T_1 分派给 M_5 , 则在 TAP 中加入一条分派顺序: $M_2[T_1] \rightarrow M_5$, 同时删除复选线 $M_5 \rightarrow (m)[M_2, M_3, M_6]$, 并改变争用线为: $[M_1] \rightarrow (z)[M_3, M_5]$;

4) 重新生成 E-TAAG 如图 3.20(b)所示, 其中 $M = [M_2, M_3, M_5]$, 将 $[M_2, M_5]$ 中的刀具 T_3 分派给 M_4 后, 其既可加工下道工序, 故在刀具分派方案表 TAP 中加入一条刀具分派顺序: $M_2[T_3] \rightarrow M_4, M_5[T_3] \rightarrow M_4$;

5) 重新生成新的 E-TAAG 如图 3.20(c)所示, $M = [M_3, M_4, M_5]$, M_1 获得相关刀具后可以进行下道工序的加工, 设 M_3 中的刀具分派给 M_1 , TAP 中加入两条分派顺序: $M_3[T_1] \rightarrow M_1, M_4[T_2] \rightarrow M_1$;

6) 重新生成 E-TAAG 如图 3.20(d)所示, 各机床节点均无刀具申请线, 刀具分派结束。

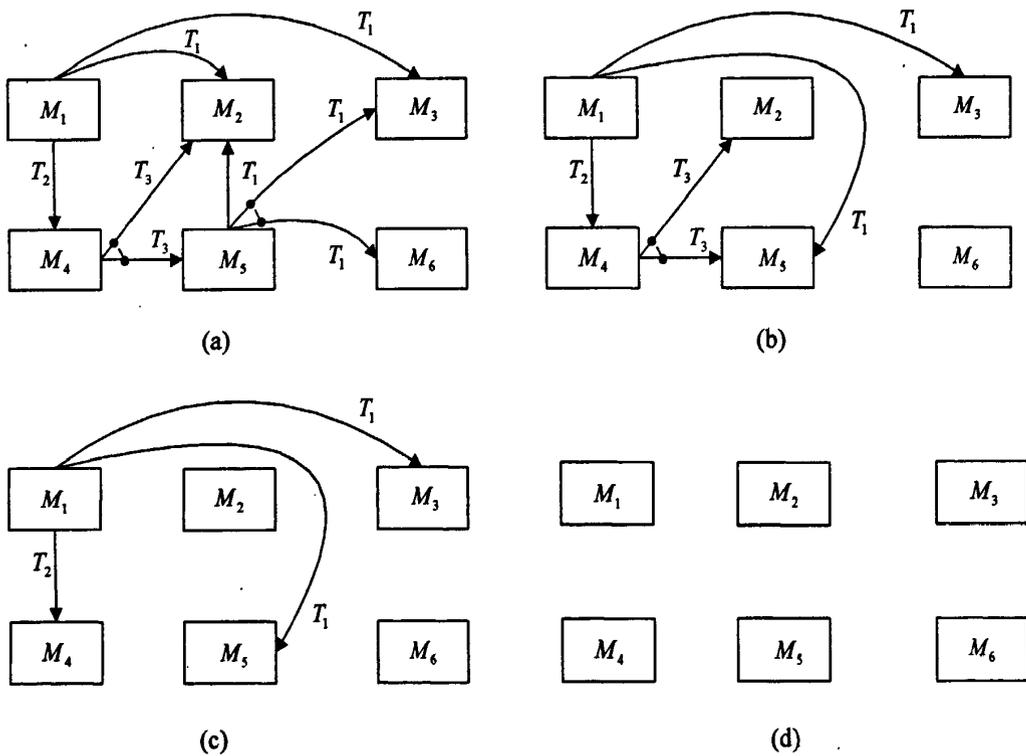


图 3.20 刀具分派过程示意图

Fig.3.20 Schematic diagram in the process of tools assignment

7) 输出刀具分派方案表 TAP 如下:

$$M_6[T_4] \rightarrow M_2$$

$$M_2[T_1] \rightarrow M_5$$

$$M_2[T_3] \rightarrow M_4, M_5[T_3] \rightarrow M_4,$$

$$M_3[T_1] \rightarrow M_1, M_4[T_2] \rightarrow M_1$$

8) 结束。

3.6 本章小结

死锁问题是刀具流研究的核心内容和主要难点, 本文在回顾和分析刀具流无死锁调度工作基础上, 将刀具寿命考虑进来, 对该问题作了进一步的分析。首先在 TAAG 基础上通过引入全选线建立了扩展的刀具申请分配图 E-TAAG, 使得其能够描述考虑刀具寿命的刀具申请分配状态, 然后在此基础上通过对刀具流死锁性质的分析建立了刀具流死锁判定定理和非死锁图的性质定理, 它们是系统两层死锁避免策略的基础, 前者用于 FMS 实现机床对不会导致死锁的下道工序的决策, 后者用于 FMS 实现刀具的无死锁分派决策, 实例分析验证了该死锁避免策略的合理性和可行性, 最后将两层死锁避免策略与调度规则结合建立了刀具流无死锁的动态调度算法, 并给出了刀具失效的处理策略, 解决了工序备刀下的刀具流死锁避免和刀具实时分派问题。

第 4 章 基于工步备刀的刀具流无死锁调度研究

上一章采用图论方法对工序备刀情况下考虑刀具寿命的刀具流死锁问题进行了深入分析，其刀具流控制的基本前提是：待加工工序所需刀具必须全部确定后才能开始该工件的加工，否则必须等待。这种控制可以避免在机床加工过程中的停机等刀情况，能够提高机床加工效率，然而它同时也会带来不利影响，例如当前待加工工序很可能迟迟不能获得所需刀具而使昂贵的机床资源长时间处于空闲状态，此外也会造成所占用的刀具资源利用率大大降低。

事实上在实际加工过程中，尤其是在高度自动化的加工中心上，采用的是工序集中原则，各工序往往包括了很多的工步，加工时间很长，因此，如果仍然将工序不可中断作为调度的前提将会对刀具的利用率造成不利影响，而如果将工序可中断（即以工步为备刀对象）作为出发点，应当可以取得更好的效果，这一方面的研究目前还很匮乏^[66]。

由此，本章对另一类备刀前提即工步备刀情况进行了重点研究，首先建立刀具分派模型，在此基础上分析工步备刀策略下刀具流死锁的相关性质，从而建立刀具流死锁判定的相关理论，提出刀具流死锁回避策略，并最终建立“死锁避免策略+启发式规则”的无死锁调度算法。

4.1 工步备刀下刀具流的状态演变

4.1.1 占有等待的定义

工步备刀前提下刀具流死锁现象的发生是以机床占有并等待（hold and wait）为前提的，一般认为机床持续占有某把刀具是因为在后续工步中还需要使用它们，这对于减少系统中的刀具交换量是有益的。

令： a_{ji} — T_j 类刀具的一个实例分配给机床 M_i ； b_{ij} —机床 M_i 需要 T_j 类刀具的一个实例； A_a —分配给机床 M_i 的 T_j 类刀具的集合； A_b —机床 M_i 对 T_j 类刀具的需求集合； $R(a_{ji})$ —机床 M_i 最后一次需求 T_j 类刀具的序列号； $R(b_{ij})$ —机床 M_i 第一次需求 T_j 类刀具的序列号； $f_i = [T_{\delta_{i(1)}}, T_{\delta_{i(2)}}, \dots, T_{\delta_{i(n)}}]$ —机床 M_i 的刀具需求序列； $\delta_{i(k)}$ —在 f_i 中第 k 步所需的刀具类型。

又令： $R(a_{ji}) = \max[k : \delta_{i(k)} = j, k = 1, 2, \dots, |f_i|]$

$R(b_{ij}) = \min[k : \delta_{i(k)} = j, k = 1, 2, \dots, |f_i|]$

定义 1 若存在 $a_{ji} \in A_a$, $b_{ik} \in A_b$, 使得 $R(a_{ji}) > R(b_{ik})$, 则称机床 M_i 是占有并等待的。

定义 2 机床 M_i 占有的刀具集合称为机床 M_i 的 H 域，用 M_i^H 表示，且 $M_i^H = T_{\text{now}}^i \cup T_{\text{follow}}^i, T_{\text{now}}^i \cap T_{\text{follow}}^i = \emptyset$ 。

其中： T_{now}^i —当前工步和后续工步都需要的刀具集合； T_{follow}^i —当前工步的先期工步和后续工步都需要的刀具集合。

定义 3 机床 M_i 等待的刀具集合称为机床 M_i 的 W 域，用 M_i^W 表示，且 $M_i^W = T_{\text{next}}^i \cup T_{\text{follow}}^i$

其中： T_{next}^i —机床 M_i 上下一工步需要申请的刀具集合； T_{follow}^i —机床 M_i 上后续工步需要申请的刀具集合。

根据刀具流死锁的产生条件及占有等待的定义，可知刀具流死锁只与机床 H 域和 W 域中的刀具有关。

4.1.2 刀具流状态演变模型

由于系统中与刀具流死锁有关的因素通常只包括机床、刀具及机床对 H 域和 W 域刀具的操作（请求，获得和释放），所以刀具流的状态可以描述为系统中机床的 W 域和 H 域刀具的申请与占有状况。

定义 4 有向图 $G=(\varphi, \phi)$ 为刀具流状态演变模型（Tool Flow State Evolution Model, TFSEM），其中：

$\varphi=\{S_i | i=1,2,\dots,n\}$ —刀具流的可达状态集，即系统中刀具资源可能的分配状态集合且 S_n 表示刀具流的分配完成状态；

$\phi = \{(S_r, S_k)^{M_i} | S_r, S_k \in \varphi, M_i \in M\}$ —系统中机床操作线的集合；

机床的操作线 $(S_r, S_k)^{M_i}$ 是指在模型中从一个状态节点指向另一个状态节点的有向线段，也可用 $S_r \xrightarrow{M_i} S_k$ 表示。该操作线表征由于机床 M_i 在状态 S_r 对其 W 域和 H 域的刀具资源实施了获得或释放操作使得状态 S_r 演变为状态 S_k 。

系统的状态模型如图 4.1 所示，其中 $\varphi = \{S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8\}$ ，

$\phi = \{(S_1, S_2)^{M_1}, (S_2, S_3)^{M_1}, (S_3, S_8)^{M_2}, (S_2, S_6)^{M_2}, (S_6, S_8)^{M_1}, (S_1, S_3)^{M_1}, (S_6, S_8)^{M_1}, (S_1, S_3)^{M_1}, (S_1, S_4)^{M_2}, (S_4, S_6)^{M_1}, (S_6, S_8)^{M_1}, (S_4, S_7)^{M_1}, (S_7, S_8)^{M_1}\}$

定义 5 若不存在 $(S_r, S_k)^{M_i}$ 操作线，即机床 M_i 在状态 S_r 不能对其 H 域和 W 域的刀具执行释放、获得操作，则称机床 M_i 在状态 S_r 阻塞。

图 4.1 中，由于 $(S_3, S_k)^{M_1}, (S_3, S_k)^{M_2} (3 < k \leq 8)$ 操作线不存在，故机床 M_1 和 M_2 在状态 S_3 阻塞。

定义 6 若机床 M_i 在状态 S_r 以及它可能到达的所有未来状态都是阻塞的，则称机床 M_i 在状态 S_r 为死锁机床，否则称为非死锁机床。

根据定义 6，图 4.1 中的机床 M_1 和 M_2 在状态 S_3 都是死锁机床。

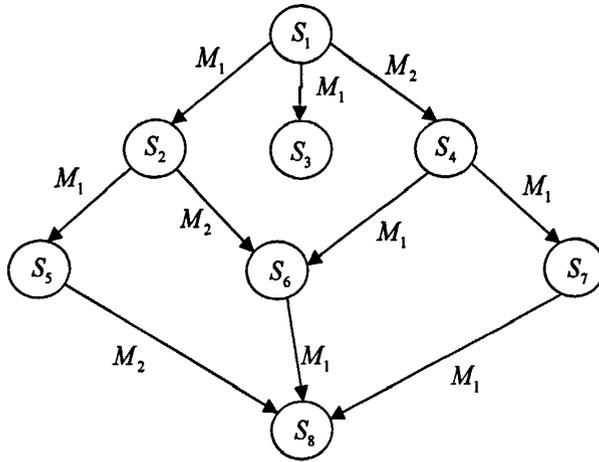


图 4.1 刀具流状态演变模型

Fig.4.1 Tool flow state evolution model

机床在某个状态 S_r 阻塞, 说明机床是占有并等待的, 机床在某个状态 S_r 死锁, 说明机床出现了循环等待的状况, 故根据刀具流死锁产生的条件, 可以给出定义 7。

定义 7 若在状态 S_r 有一个死锁机床 M_i , 则称状态 S_r 为死锁状态(Deadlock state)。

定义 8 设 $\beta = \{S_r | r \in \{1, 2, \dots, n\}\}$ 为刀具流的可达状态集 φ 的一个子集, 其中 S_r 演变为 S_{r+k} ($k > 0$) 是由于某个刀具分派所引发的, 若状态 S_{r+k} 不发生死锁, 至少存在一个 $S_{r+k+\lambda}$ ($r+k+\lambda < n, \lambda > 0$) 死锁, 则称状态 S_r 为无死锁不安全状态(Deadlock-free unsafe state)。

定义 9 设 $\beta = \{S_r | r \in \{1, 2, \dots, n\}\}$ 为刀具流的可达状态集 φ 的一个子集, 对 $\forall S_r \in \beta$, 都存在一个刀具分派顺序, 使得系统由 S_r 演变为 S_n , 则称状态 S_r 为安全状态(safe state)。

综上所述, 刀具流的状态是随着机床对其 H 域和 W 域的刀具的操作而变化的, 只要机床未阻塞, 它就能改变系统的状态, 而且在刀具流状态演变模型中的每个状态, 都代表了系统中刀具资源的分配状况, 该状况可进一步用刀具资源分配图 TRAG 来描述。

4.1.3 刀具资源分派图

为了描述系统中某状态下的刀具资源分派状况, 建立刀具资源分派图(Tool Resource Allocation Graph, TRAG)如下:

定义 10 $TRAG = G(V_M, V_T, E_A, E_B, S_{AB}, S_{BA})$,

$V_M = \{M_1, M_2, \dots, M_n\}$ 为机床顶点集合;

$V_T = \{T_1, T_2, \dots, T_m\}$ 为刀具顶点集合;

$E_A = \{a_{ji}^\sigma | a_{ji}^\sigma = T_j \rightarrow M_i, i, j, \sigma = 1, 2, \dots\}$ 为刀具分派线集合, $T_j \rightarrow M_i$ 为从刀具 T_j 到机床 M_i 的有向线段, 表示 T_j 类刀具实例在机床 M_i 的 H 域, σ 为 T_j 类刀具实例号;

$E_B = \{b_{ij} | b_{ij} = M_i \rightarrow T_j, i, j = 1, 2, \dots\}$ 为刀具申请线集合, $M_i \rightarrow T_j$ 为从机床 M_i 到 T_j 类刀具的有向线段, 表示机床 M_i 的 W 域的刀具申请;

$S_{AB} = \{R(a_{ji}^\sigma)l_{ji}^\sigma | i, j, \sigma = 1, 2, \dots\}$ 分派线的权集合, 其中 $R(a_{ji}^\sigma) \in Z^+$ 为 T_j 类刀具的第 σ 把刀具在机床 M_i 上的释放步骤, $l_{ji}^\sigma \in R^+$ 为第 σ 把刀具在机床 M_i 上的使用时间;

$S_{BA} = \{(R_k(b_{ij}), l_{ij}^k) | i, j, k = 1, 2, \dots\}$ 申请线的权集合, $R_k(b_{ij}) \in Z^+$ 为机床 M_i 第 k 次需求刀具 T_j 的步骤, $l_{ij}^k \in R^+$ 为与 $R_k(b_{ij})$ 相应的使用时间;

以矩形表示各类刀具区域, 矩形内圆圈表示各类刀具资源的实例, 其中的数字表示刀具的剩余寿命, 圆形表示机床, 在矩形和圆形之间加入各刀具申请线和刀具分派线并标识相应的权, 即形成了刀具资源分派图 TRAG, 如图 4.2 所示。

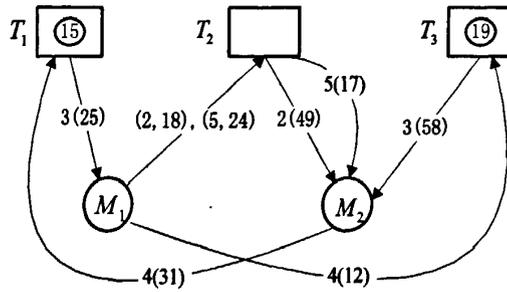


图 4.2 刀具资源分派图

Fig.4.2 Tool resource allocation graph

图 4.2 中各机床刀具申请线和刀具分派线的权的具体含义如下:

$R(a_{11}^1) = 3, l_{11}^1 = 25$; $R_1(b_{12}) = 2, l_{12}^1 = 18$; $R_2(b_{12}) = 5, l_{12}^2 = 24$; $R_1(b_{13}) = 4, l_{13}^1 = 12$;
 $R(a_{22}^1) = 2, l_{22}^1 = 49$; $R(a_{22}^2) = 5, l_{22}^2 = 17$; $R(a_{32}^1) = 3, l_{32}^1 = 58$; $R_1(b_{21}) = 4, l_{21}^1 = 31$ 。

由此可见刀具资源分派图 TRAG 清晰的表达了系统中某时刻刀具流的状态, 主要包括各机床的 W 域的刀具需求情况 (需求步骤和需求寿命)、H 域的刀具的占用情况 (释放步骤和被消耗掉的寿命) 及未分派的刀具的剩余寿命情况。

基于 TRAG, 给出如下定义:

定义 11 若 TRAG 中, 存在刀具申请线集 $\{(M_1 \rightarrow T_1), (T_1 \rightarrow M_2), \dots, (M_{n-1} \rightarrow T_{n-1}), (T_{n-1} \rightarrow M_n)\}$, 且 $M_1 = M_n$, 则称 TRAG 中存在环路 (Bound Circle, BC)。

定义 12 若 TRAG 中, 只有孤立的节点, 则称该 TRAG 为可完全化简 TRAG, 否则称为不可完全化简 TRAG。

4.2 刀具流死锁状态的判定

4.2.1 刀具流死锁判定定理

首先给出“刀具满足机床申请”和“关键刀具”的定义。

定义 13 若 T_j 类未分派刀具的剩余寿命能够完成机床 M_i 使用此类刀具的各分工

步的加工, 则称 T_j 类未分派刀具满足机床 M_i 的申请。表示为 $L_{\sigma}(T_j) \ominus L_i(T_j)$, 否则表示为 $L_{\sigma}(T_j) \bar{\ominus} L_i(T_j)$ 。其中: $L_{\sigma}(T_j) = \{l_{\sigma}^k(T_j) | k=1,2,\dots,\sigma\}$ 为 T_j 类未分派刀具的剩余寿命集合; $l_{\sigma}^k(T_j)$ 为 T_j 类刀具实例的剩余寿命; $L_i(T_j) = \{l_{ij}^k | k=1,2,\dots,\delta\}$ 为机床 M_i 各分工步使用 T_j 类刀具的时间集合, l_{ij}^k 为机床 M_i 使用 T_j 类刀具的各分工步的加工时间。

定义 14 设机床 M_i 的最大释放步骤 $max(M_i) = \max\{R(a_{ji}^{\sigma}) : a_{ji}^{\sigma} \in E_A\}$, 最小需求步骤 $min(M_i) = \min\{R_k(b_{ij}) : b_{ij} \in E_B\}$, $L_i^*(T_j) = \{l_{ij}^k : R_k(b_{ij}) < max(M_i), k \in \varphi\}$, $\varphi = 1,2,\dots,\delta$, 若 T_j 类刀具满足如下条件:

- 1) $L_{\sigma}(T_j) \bar{\ominus} L_i^*(T_j)$;
- 2) $\forall M_i$, 有 $a_{ji}^{\sigma} \in E_A : R(a_{ji}^{\sigma}) > min(M_i)$;
- 3) $\exists M_i$, 有 $b_{ij} \in E_B : \min\{R_k(b_{ij})\} < max(M_i)$ 。

则称 T_j 类刀具为关键刀具。

条件 1) 说明可获得的 T_j 类刀具的寿命不满足机床 M_i 的部分工步或全部工步的申请; 条件 2) 说明 T_j 类刀具的实例都被机床占用; 条件 3) 说明存在机床 M_i 需要获得 T_j 类刀具实例后才能释放其占用的刀具。

定理 1 刀具流发生死锁的充分必要条件是当且仅当 TRAG 中存在一个强连通图 K , 且 $\forall T_j \in K$ 为关键刀具。

证明:

充分性: 若 TRAG 中存在强连通图 K , 则 K 中任意两个节点都是相互可达的, 故必可做一回路 BC , 且 BC 必经过 K 中所有各点, 若不然, 则必有一回路不包含某一节点 v , 并且 v 与回路 BC 上的各节点就不是相互可达的, 与强连通图条件矛盾。据已知条件: $\forall T_j \in K$ 为关键刀具, 则有 $\forall T_j \in BC$ 为关键刀具, 根据关键刀具的定义可知, $\forall M_i \in BC$ 其所需要的刀具都被 BC 中其它机床占用, 既 BC 上的机床发生了循环等刀现象, 根据死锁的条件可知, 刀具流发生了死锁。

必要性: 假设刀具流死锁, 则死锁中的机床 M_i 至少需要 1 把 T_j 类刀具, 而此刀具被死锁中的另一个机床占用, 构建一个机床等待图如图 4.3 所示, 节点是死锁中的机床, 弧为死锁中机床间的关系, 也即 $\langle M_i, M_k \rangle$ 存在当且仅当 T_j 类可得刀具的寿命不满足 M_i 的申请, 而 T_j 类刀具的其它实例又被机床 M_k 占用。

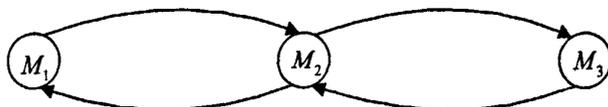


图 4.3 机床等待图

Fig.4.3 Machine-wait graph

因为有一个有限的机床节点集合，集合中每个机床节点都至少有一个输出弧指向集合中的其它机床节点，这也暗示着机床等待图中存在一格回路 BC，在回路 BC 中插入刀具节点如图 4.4 所示，构成新的回路 BC'，也即刀具流发生死锁，则 TRAG 中一定存在回路 BC'，且 BC' 至少包含各刀具节点与机床节点一次，即 BC' 中任意两个节点都是相互可达的，也即 TRAG 中一定存在一个强连通图，称为 K。

现假设 $\forall T_j \in K$ 不是关键刀具，若关键刀具定义的条件 1) 不成立，即机床 M_i 对 T_j 类刀具的寿命申请满足，这与机床等待图的构建矛盾，如果条件 2) 不成立，即 K 中至少有一个机床，其 T_j 类刀具实例不用进一步的分派即可释放，这也与机床等待图的构建矛盾，如果条件 3) 不成立，即 $\exists M_i \in K$ ，不需要分派 T_j 类刀具实例即可释放其占有的刀具，这与 K 中所有机床节点构成强连通图矛盾，故 $\forall T_j \in K$ 为关键刀具，命题得证。

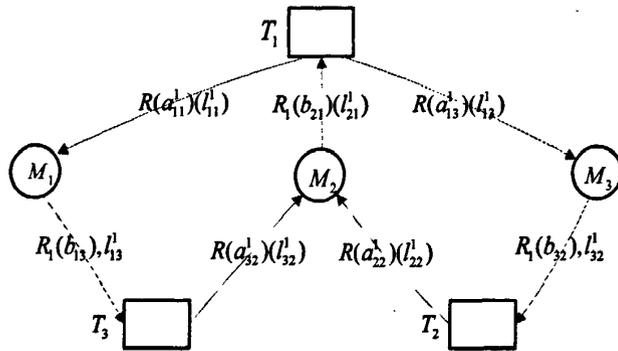


图 4.4 扩展的机床等待图

Fig.4.4 Extended machine-wait graph

4.2.2 刀具流死锁检测算法

基于定理 1 构建刀具流死锁检测算法流程如图 4.5 所示。

对于一个给定的刀具资源分派图 TRAG，首先确定只有刀具申请线（或刀具分派线）的节点，然后删除与此节点相关的刀具申请线（或刀具分派线），上述操作重复进行，直到没有这样的节点出现，也即说明刀具资源分派图 TRAG 中不存在刀具申请线和刀具分派线或者存在的节点都有刀具申请线和刀具分派线，如果 TRAG 中不存在刀具申请线或刀具分派线，则刀具流不发生死锁。如果有刀具申请线和刀具分派线同时存在的节点，则判断关键刀具的存在，即每把刀具都应服从定义 14 中的关键刀具的条件，如果不满足，则存在非关键刀具，删除非关键刀具节点相关的刀具申请线（或刀具分派线），并重复上述操作，如果存在满足关键刀具条件的节点，则刀具流发生死锁。

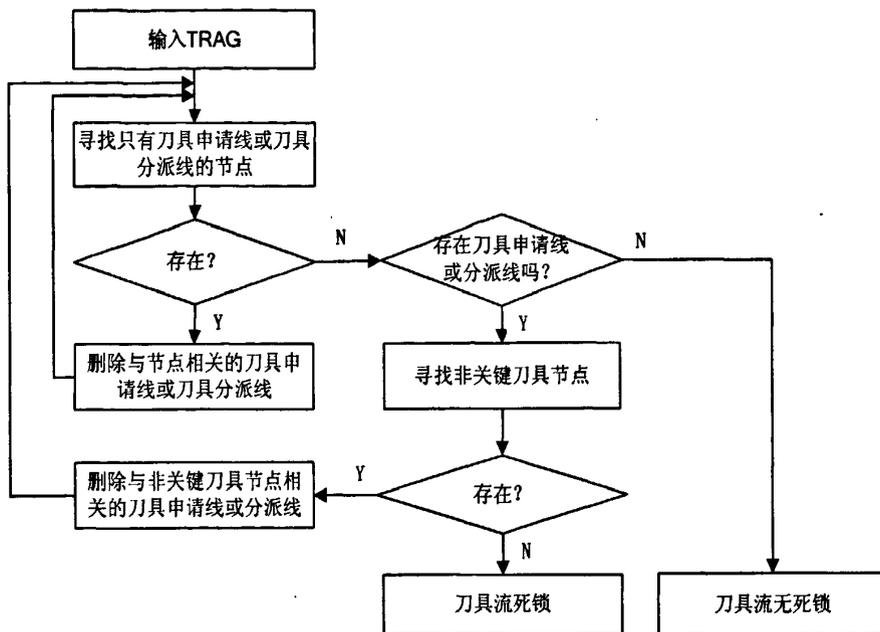


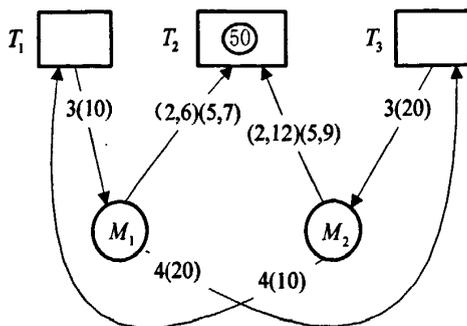
图 4.5 刀具流死锁检测算法流程图

Fig.4.5 Flow chart of tool flow deadlock detection algorithm

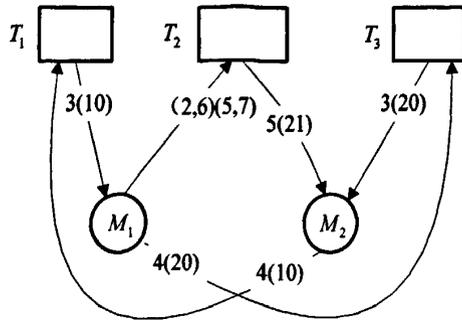
死锁检测算法可用以检测某个刀具的分派是否会引发死锁，进而决定该刀具是否应当分派，因此该方法是利用单步前瞻策略进行死锁避免。但是当系统中存在无死锁不安全状态时，该方法将不起作用，因此必须进一步对无死锁不安全状态的检测进行分析。

4.3 刀具流无死锁不安全状态的判定

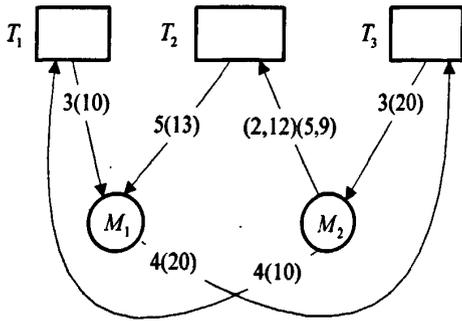
首先给出无死锁不安全状态示例如图 4.6 所示。图 4.6(a)中每个机床都获得了初始加工所需要的刀具 (T_1 分派给 M_1 , T_3 分派给 M_2), 且没有死锁发生(没有关键刀具)。机床 M_1 和 M_2 都在请求下一步加工需要的 T_2 刀具, 如果 T_2 刀具分派给了 M_2 , 则出现了如图 4.6(b)的情况, 如果 T_2 刀具分派给了 M_1 , 则出现了如图 4.6(c)的情况。



(a)



(b)



(c)

图 4.6 无死锁不安全状态及其导致的死锁状态

Fig.4.6 Deadlock-free unsafe state and resulting deadlock state

也即刀具 T_2 无论分派给哪个机床都发生死锁。所以图 4.6(a)所示的情况为无死锁不安全状态，也即在此状态，不存在一个刀具分派顺序使系统中的工件都顺利得到加工。

4.3.1 机床非死锁判定定理

如果某时刻系统状态为无死锁不安全状态，则在此状态，一定不存在一个刀具分派顺序使系统中的工件都顺利得到加工，即系统中至少存在一个机床，其在将来所有状态都不能获得其加工所需的刀具，也即不能够执行获得刀具的操作，根据定义 6，该机床为死锁机床，所以系统的无死锁不安全状态等价于系统中存在死锁机床，故对无死锁不安全状态检测时只需检测死锁机床的存在，故需要建立机床非死锁判定定理，另外为了减少刀具流无死锁不安全状态检测的负荷，根据定义 5 给出了相应的 TRAG 简化规则。

(1) 机床非死锁判定定理

根据定义 6 中机床非死锁的定义，给出其判定定理。

定理 2 TRAG 中，若机床 M_i 对 $\forall T_j, j=1,2,\dots$ ，满足： $L_{av}(T_j) \otimes L_i(T_j)$ ，则机床 M_i

是非死锁的。其中： $L_{\sigma}(T_j) = \{l_{\sigma}^k(T_j) | k=1,2,\dots,\sigma\}$ 为刀具节点 T_j 的可得刀具剩余寿命集合， $L_i(T_j) = \{l_y^k | k=1,2,\dots,\delta\}$ 为机床 M_i 各分工步使用 T_j 类刀具的时间集合。

证明：

充分性：若 $\forall T_j, j=1,2,\dots$ 有 $L_{\sigma}(T_j) \odot L_i(T_j)$ 成立，则机床 M_i 对任意刀具资源 T_j 的请求都可以满足，机床 M_i 可以执行获得 $T_j, j=1,2,\dots$ 实例的操作，即机床 M_i 在当前状态以及它可能到达的所有未来状态都是非阻塞的，由定义 6 知机床 M_i 是非死锁的。

必要性：若机床 M_i 是非死锁机床，那么其在当前状态以及它可能到达的所有未来状态都是非阻塞的，也即机床 M_i 对任意刀具 $T_j, j=1,2,\dots$ 的请求应该可以满足，显然有 $L_{\sigma}(T_j) \odot L_i(T_j)$ 成立。

(2) TRAG 简化规则

1) 简化规则 1 TRAG 中，存在机床节点 M_i ，其刀具申请线的权集合 $S_{BA} = \{(R_k(b_{ij}), l_{ij}^k) | j,k=1,2,\dots\}$ ，刀具分派线的权集合为 $S_{AB} = \{R(a_{ji}^{\sigma})(l_{ji}^{\sigma}) | j,\sigma=1,2,\dots\}$ ，若 $\exists R(a_{ji}^{\sigma})(l_{ji}^{\sigma}) \in S_{AB}$ 满足： $\forall R_k(b_{ij}) > R(a_{ji}^{\sigma}), f=1,2,\dots$ ，且 $f \neq j$ ， $R_k(b_{if}) \in S_{BA}$ ，则拥有权 $R(a_{ji}^{\sigma})(l_{ji}^{\sigma})$ 的刀具分派线可以删除，并更新相应刀具的寿命。

简化规则 1 的基本思想：若有 $\forall R_k(b_{if}) > R(a_{ji}^{\sigma}), f=1,2,\dots$ ，且 $f \neq j$ ，说明机床 M_i 申请刀具 T_f 但并不占有刀具 T_j ，也即刀具 T_j 用完后即可释放，故拥有权 $R(a_{ji}^{\sigma})(l_{ji}^{\sigma})$ 的刀具分派线可以删除。简化规则 1 示例如图 4.7 所示。



图 4.7 简化规则 1 示例

Fig.4.7 Example of reduction rules 1

2) 简化规则 2 TRAG 中，若存在刀具节点 T_j ，可得刀具的剩余寿命集合为 $L_{\sigma}(T_j) = \{l_{\sigma}^k(T_j) | k=1,2,\dots,\sigma\}$ ，刀具申请线的权集合为 $S_{BA} = \{(R_k(b_{ij}), l_{ij}^k) | i,k=1,2,\dots\}$ ， $L_i(T_j) = \{l_y^k | k=1,2,\dots,\delta\}$ 为机床 M_i 各分工步对 T_j 类刀具的申请，若 $L_{\sigma}(T_j) \odot L_i(T_j)$ ，则可以删除刀具申请线 $b_{ij} = M_i \rightarrow T_j$ ，同时更新集合 $L_{\sigma}(T_j)$ 。

简化规则 2 的基本思想：若表达式 $L_{\sigma}(T_j) \odot L_i(T_j)$ 成立，则 T_j 可得刀具的剩余寿命满足机床 M_i 的申请，也即机床 M_i 可以执行获得刀具 T_j 的操作，故可以删除机床 M_i 指向刀具节点 T_j 的刀具申请线。简化规则 2 示例如图 4.8 所示。

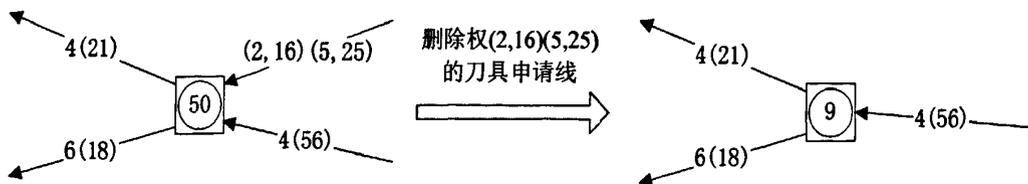


图 4.8 简化规则 2 示例

Fig.4.8 Example of reduction rules 2

3) 简化规则 3 TRAG 中, 若存在刀具节点 T_j , 其可得刀具的剩余寿命集合为 $L_{av}(T_j) = \{l_{av}^k(T_j) | k=1,2,\dots,\sigma\}$, 刀具申请线集合为 $E_B = \{b_{ij} | b_{ij} = M_i \rightarrow T_j, i=1,2,\dots\}$, 其权集合为 $S_{BA} = \{(R_k(b_{ij}), l_{ij}^k) | i,k=1,2,\dots\}$, 设 $L_i^*(T_j) = \{l_{ij}^k | k=1,2,\dots\}, i=1,2,\dots$, 若 $\exists b_{ij} \in E_B$ 其起始点机床节点 M_i 有且仅有此一条申请线, 同时满足 $L_{av}(T_j) \odot L_i^*(T_j)$, 则可以删除 M_i 相关的刀具申请线和刀具分派线, 同时更新集合 $L_{av}(T_j)$ 。

简化规则 3 的基本思想: 机床 M_i 有且仅有此一条刀具申请线, 且 T_j 的可得刀具剩余寿命满足其申请, 则机床 M_i 获得 T_j 类刀具后即可释放其占有的所有刀具, 也即 M_i 一定是非死锁机床, 故可删除与其相关的刀具申请线和刀具分派线。简化规则 3 示例如图 4.9 所示。

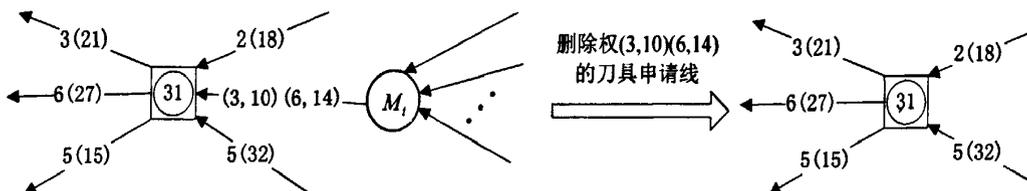


图 4.9 简化规则 3 示例

Fig.4.9 Example of reduction rules 3

4) 简化规则 4 TRAG 中, 若存在刀具节点 T_j , 刀具申请线的权集合为 $S_{BA} = \{(R_k(b_{ij}), l_{ij}^k) | i,k=1,2,\dots\}$, 分派线的权集合为 $S_{AB} = \{R(a_{ji}^\sigma)(l_{ji}^\sigma) | i,\sigma=1,2,\dots\}$, $L_{av}^i(T_j)$ 为 T_j 类被机床 M_i 占用的刀具实例的剩余寿命, 若 $L_{av}^i(T_j) < \min\{l_{ij}^k | i,k=1,2,\dots\}$, 则删除此 T_j 类实例指向机床 M_i 的刀具分派线。

简化规则 4 的基本思想: $L_{av}^i(T_j) < \min\{l_{ij}^k | i,k=1,2,\dots\}$ 说明此把刀具不满足任意机床加工的需要, 故可以删除此刀具实例的分派线。

5) 简化规则 5 TRAG 中, 设机床 M_i 的刀具申请线的权集合 $S_{BA} = \{(R_k(b_{ij}), l_{ij}^k) | j,k=1,2,\dots\}$, 刀具分派线的权集合为 $S_{AB} = \{R(a_{ji}^\sigma)(l_{ji}^\sigma) | j,\sigma=1,2,\dots\}$, $L_i^*(T_j) = \{l_{ij}^k | l_{ij}^k: R_k(b_{ij}) < R(a_{ji}^\sigma), j \neq i, \sigma=1,2,\dots\}$ 为机床 M_i 各分工步使用 T_j 类刀具的时间集合, 刀具节点 T_j 的可得刀具剩余寿命集合为 $L_{av}(T_j) = \{l_{av}^k(T_j) | k=1,2,\dots,\sigma\}$, 若 $L_i^*(T_j) \neq \emptyset$ 且 $\exists T_j$ 满足: $L_{av}(T_j) \odot L_i^*(T_j)$, 则可以删除拥有权 $R(a_{ji}^\sigma)(l_{ji}^\sigma)$ 的刀具分派线, 并更新相应刀具寿命。

简化规则 5 的基本思想是： $L_i^*(T_j) \neq \emptyset$ ，则存在机床 M_i 占有刀具 T_k 申请刀具 T_j ，若同时 T_j 类可得刀具的剩余寿命满足： $L_{av}(T_j) \ominus L_i^*(T_j)$ ，说明集合 $L_i^*(T_j)$ 中工步的刀具申请能够得到满足，也即刀具 T_k 能够释放。简化规则 5 示例如图 4.10 所示。

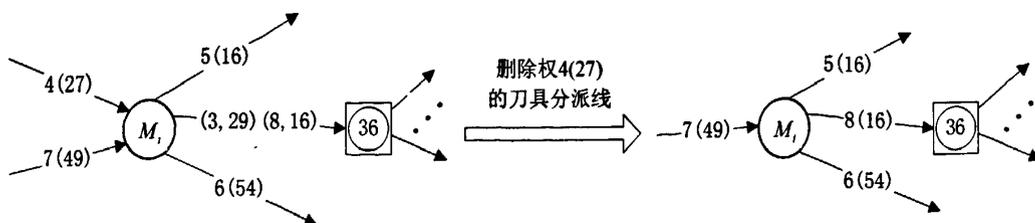


图 4.10 简化规则 5 示例

Fig.4.10 Example of reduction rules 5

6) 简化规则 6 TRAG 中，若存在刀具节点 T_j ，只有刀具分派线，则删除与刀具节点 T_j 相关的刀具分派线。

7) 简化规则 7 TRAG 中，若存在机床节点 M_i ，只有刀具分派线，则删除与机床节点 M_i 相关的刀具分派线。

4.3.2 无死锁不安全状态检测算法

基于机床非死锁判定定理和 TRAG 简化规则建立刀具流无死锁不安全状态检测算法流程如图 4.11 所示。

Step1: 根据系统某时刻刀具的申请分派状态，建立刀具资源分派图 TRAG;

Step2: 根据给定的简化规则，对 TRAG 进行简化;

Step3: 简化之后的 TRAG 若存在具有未分派刀具的刀具争用节点 $N_k, k=1,2,\dots$ ，则转 step4，否则转 step10;

Step4: 确定各刀具争用节点 N_k 的刀具分派方案集 $F(N_k) = \{f_i(N_k) | i=1,2,\dots\}$ ， $F(N_k), k=1,2,\dots$ 中各 $f_i(N_k)$ 自由组合成分派方案组合集 FAN;

Step5: 为了确定分派方案组合集 FAN 中各刀具分派方案组合的可行性，从 FAN 中选择一种刀具分派方案组合进行试分派，为此需要建立虚拟的 TRAG，用 V-TRAG 表示;

Step6: 根据给定的简化规则，对 V-TRAG 进行简化;

Step7: 若 V-TRAG 可完全化简，说明选定的刀具分派方案组合可行，系统为安全状态，否则 step8;

Step8: 若 V-TRAG 不可完全化简，且分派方案组合集 FAN 不为空，说明还有其他分派方案组合，则需重新试分派各争用节点 N_k 的其他分派方案，并继续 step5-step6，否则 step9;

Step9: 若分派方案组合集为空, 说明所有分派方案组合都已经试分派过, 也即在刀具争用节点 N_k 不存在可行的分派方案使系统从初始状态演变为最终状态, 故 TRAG 所代表的状态为无死锁不安全状态。

Step10: 若 TRAG 为可完全化简, 则某时刻系统为安全状态, 否则为无死锁不安全状态。

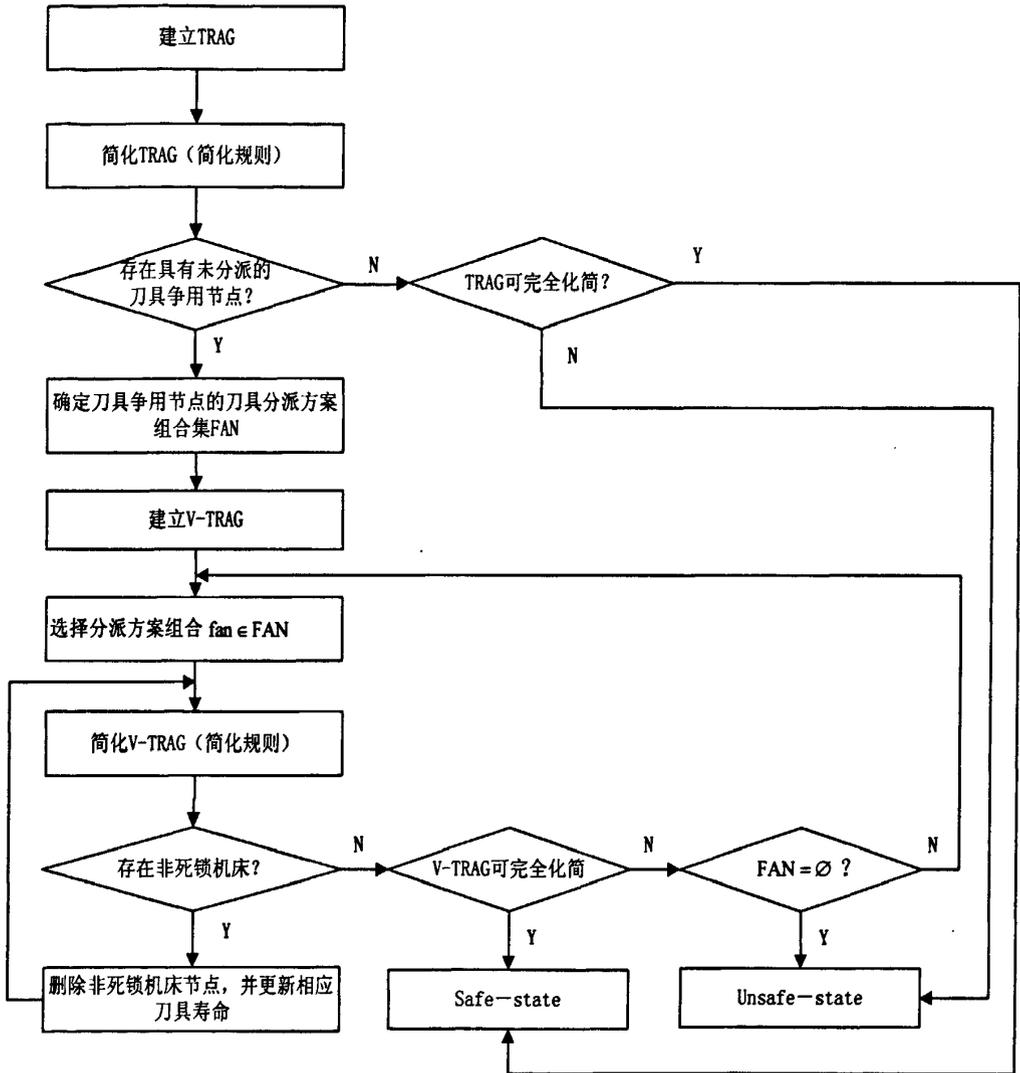


图 4.11 无死锁不安全状态检测算法流程图

Fig.4.11 Flow chart of deadlock-free unsafe state detection algorithm

刀具流无死锁不安全状态检测算法可用以检测某个刀具的分派是否会引发无死锁不安全状态, 也即判定某把刀具分派后, 是否存在一个刀具分派顺序使系统中的工件都能顺利得到加工, 进而决定该刀具是否应当分派。它为后续刀具流无死锁调度算法的顺利实施提供了技术支撑。

4.4 刀具流无死锁调度算法的实现

4.4.1 死锁避免策略的基本思想

刀具流死锁避免策略应该保证系统在机床选择工件和刀具选择机床时所形成的 TRAG 不能存在死锁状态和无死锁不安全状态，针对此情况，给出相应的两层死锁避免策略如图 4.12 所示。

当机床决策点到来时，可决策机床在各自的虚拟工序队列中选取下道工序，根据所选工序的工艺信息及当前的刀具分派状态即可确定所选工序的刀具需求建立刀具资源分派图 TRAG，然后进行死锁状态和无死锁不安全状态的判定，从中选择不会导致死锁及无死锁不安全状态的的下道工序，此即为一层死锁避免。它保证一定存在着可行的刀具分派方案使得各机床的下道工序都能顺利地得到加工。当刀具决策点到达时，为了保证在现有刀具资源下每次刀具的分派都能使机床的后续工步得到加工，应该先建立虚拟的 TRAG 以进行刀具的试分派，在保证此次刀具分派一定会使系统处于安全状态下，才能进行实际的刀具分派，从而保证了系统一定存在可行的刀具分派顺序完成加工操作，此为二层死锁避免。

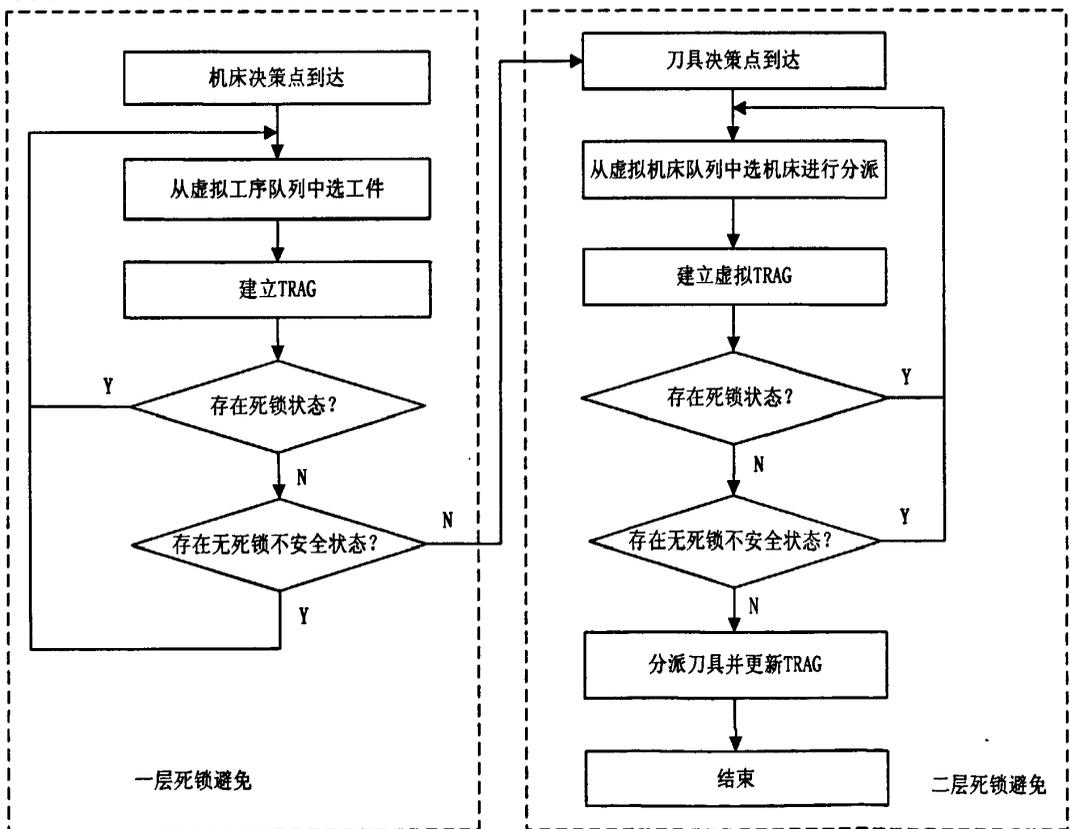


图 4.12 一般刀具需求序列的死锁避免策略

Fig.4.12 Deadlock avoidance strategy of general tool requirement sequence

4.4.2 特殊刀具需求序列的死锁避免策略

当机床的刀具需求序列呈现某些特殊结构的时候, 刀具流只有死锁状态和安全状态, 不存在无死锁不安全状态, 故刀具流死锁避免策略只需保证系统在机床选择工件和刀具选择机床时所形成的 TRAG 不存在死锁状态即可。

具体讨论之前, 首先给出相关的定义及其形式化描述。

定义 15 至少在一个机床的刀具需求顺序中重复出现的刀具集合称为重复刀具集, 其形式化描述为: $T_{RM} = \{T_j | \exists f_i \text{ with } R(b_j) < R(a_{ji})\}$ 。

定义 16 在任何机床的刀具需求顺序中都未重复出现的刀具集合称为非重复刀具集, 其形式化描述为: $T_{NRM} = \{T_j | \forall f_i \text{ with } T_j \in f_i, R(b_j) = R(a_{ji})\}$ 。

定义 17 设 $T_j \in (f_i \cap T_{RM})$, 则 f_i 中第一次出现 T_j 与最后一次出现 T_j 之间的刀具集合称为 T_j 在 f_i 中的包含刀具集 sc_{ji} , 其形式化描述为:

$$sc_{ji} = \{T_k | T_k \in f_i \text{ and } R(b_j) < R(b_k) < R(a_{ji})\}。$$

比如: 机床 M_1 的刀具需求顺序 $f_1 = (T_4, T_5, T_1, T_4, T_3)$ 中, $T_{RM} = \{T_4\}$, $T_{RM} \cap f_1 = T_4$, 则刀具 T_4 的包含刀具集 $sc_{41} = \{T_5, T_1\}$ 。

定理 3 工步备刀前提下, $T_j \in T_{NRM}$ 不是关键刀具。

证明: 假定 $T_j \in T_{NRM}$, T_j 类刀具的实例分派给了机床 M_i , 且 T_j 在 TRAG 中是关键刀具, 则根据定义 12 的条件 3) 存在刀具 T_k 有: $R(a_{ji}) > R(b_k)$, 因为基于工步备刀有: $R(b_j) < R(b_k)$; 故: $R(b_j) < R(a_{ji})$, 这与 $T_j \in T_{NRM}$ 有: $R(b_j) = R(a_{ji})$ 矛盾, 故 $T_j \in T_{NRM}$ 不是关键刀具。

定理 4 工步备刀前提下, 设加工过程的机床集合为 M , 若 $\forall M_i \in M$ 其刀具需求顺序 f_i 满足: 至多有一把刀具 $T_j \in f_i$ 属于重复刀具集 T_{RM} , 则系统为安全状态。

证明: 假定对 $\forall M_i \in M$, 其刀具需求序列中至多有一把刀具 $T_j \in T_{RM}$, 系统发生死锁状态或无死锁不安全状态, 则 TRAG 中一定会出现死锁结构 K , 设 $M_i \in K$, 则 M_i 的需求序列中至少有两把刀具处于死锁状态, 也即机床 M_i 至少占有刀具 $T_j \in K$, 等待刀具 $T_k \in K$, 即 T_j 和 T_k 都是关键刀具, 根据定理 1 有 $T_j, T_k \in T_{RM}$, 这与已知条件矛盾, 故 TRAG 中不会出现死锁结构, 系统为安全状态。

基于定理 4, 给出示例如表 4.1 所示。在工步备刀前提下, 表 4.1 的机床集合不会陷入死锁状态, 因为集合中每个机床的刀具需求顺序中至多有一把刀具属于重复刀具。

定理 5 设加工过程的机床集合为 M , 重复刀具集为 T_{RM} , 若 $\forall T_j \in T_{RM}$ 满足: $\{\bigcup_{i=1} sc_{ji}\} \cap T_{RM} = \emptyset$, 则系统处于安全状态。

证明: 假定题设条件下, 系统处于死锁状态, 则 TRAG 中一定存在死锁结构 K , 设 $M_i \in K$, 则 M_i 的需求序列中至少有两把刀具处于死锁状态, 也即机床 M_i 至少占有

刀具 $T_j \in K$ ，等待刀具 $T_k \in K$ ，即 T_j 和 T_k 都是关键刀具，也即 $\{T_j, T_k\} \subseteq T_{RM}$ ，因为 $sc_{ji} \cap T_{RM} = \emptyset$ ，故有 $T_k \notin sc_{ji}$ ，因而有： $R(a_{ji}) < R(b_{ik})$ ，这与 T_j 是关键刀具矛盾，故 TRAG 中不存在死锁结构，系统处于安全状态。

表4.1 定理4示例

Tab.4.1 Example of theorem 4

机床	刀具需求顺序	重复刀具
M_1	$(T_2, T_3, T_4, T_2, T_1, T_2)$	T_2
M_2	$(T_7, T_3, T_4, T_5, T_7, T_1)$	T_7
M_3	$(T_6, T_4, T_3, T_1, T_6, T_5)$	T_6
M_4	$(T_4, T_7, T_1, T_5, T_3, T_7)$	T_7

基于定理 5，给出示例，假定根据工件流确定了机床的加工工序，则相应的刀具需求顺序如表 4.2 所示。

则 $T_{RM} = \{T_5, T_4, T_7\}$ ， $T_{NRM} = \{T_2, T_3, T_1, T_6, T_8\}$ ， $sc_{51} = \{T_3, T_1\}$ ， $sc_{52} = \{T_6, T_2\}$ ， $sc_{55} = \{T_8\}$ ， $sc_{42} = \{T_1, T_3\}$ ， $sc_{45} = \emptyset$ ， $sc_{73} = \{T_2, T_3\}$ ， $sc_{75} = \{T_6, T_1\}$ ，因为任何一把重复刀具，其包含刀具集中只含有非重复刀具，根据定理 5 可知系统处于安全状态。

表4.2 定理5示例

Tab.4.2 Example of theorem 5

机床	刀具需求顺序	重复刀具
M_1	(T_5, T_3, T_1, T_5)	T_5
M_2	$(T_4, T_1, T_3, T_4, T_5, T_6, T_2, T_5)$	T_4, T_5
M_3	(T_7, T_2, T_3, T_7)	T_7
M_4	(T_4, T_2, T_3, T_8)	T_4
M_5	$(T_5, T_8, T_5, T_7, T_6, T_1, T_7)$	T_5, T_7

定理 4 和 5 说明，如果机床的刀具需求顺序出现题设中的情况，工步备刀前提即可避免死锁。

定理 6 工步备刀前提下，设加工过程的机床集合为 M ，若 $\forall M_i \in M$ 其刀具需求顺序 f_i 满足：至多有两把刀具 $t_j \in f_i$ 属于重复刀具集合 T_{RM} ，则单步前瞻即可避免死锁。

证明：假设系统处于无死锁不安全状态（任何一把刀具的分派都会导致死锁的状态），因为不是死锁状态，所以一定存在可得刀具 T_j ，且被机床 M_i 需求，因而，如果分派 T_j 给 M_i ，TRAG 一定存在一个死锁结构 K ， K 中的 M_i 占有刀具 T_j 申请刀具 $T_k \in K$ ，即 T_j, T_k 都是关键刀具因而 $\{T_j, T_k\} \subseteq T_{RM}$ 。这就暗含着在无死锁不安全状态下 M_i 没有占用刀具（因为每个机床至多有两把重复刀具，根据定理 1，每个机床的刀具需求序列中至多有两把关键刀具），也就是说在无死锁不安全状态下，任何一台没有占用刀具的机床需求一把可得刀具，任何已经占用刀具的机床需求一把不可得刀具，这就暗

含着无死锁不安全状态中包含了一个死锁状态，矛盾，故系统只能处于死锁状态。

基于定理 6，给出示例，假定根据工件流确定了机床的加工工序，则相应的刀具需求顺序如表 4.3 所示。

$T_{RM} = \{T_1, T_4, T_7\}$, $T_{NRM} = \{T_2, T_3, T_5, T_6, T_8\}$, $sc_{11} = \{T_2, T_4\}$, $sc_{12} = \{T_2, T_4, T_6\}$, $sc_{13} = \emptyset$, $sc_{14} = \{T_4\}$, $sc_{15} = \{T_8, T_7\}$, $sc_{41} = \{T_1, T_5\}$, $sc_{42} = \{T_5, T_3, T_1, T_6\}$, $sc_{43} = \emptyset$, $sc_{44} = \{T_1\}$, $sc_{45} = \emptyset$, $sc_{73} = \{T_2, T_5\}$, $sc_{75} = \{T_1, T_6, T_3\}$ ，因为 sc_{11} , sc_{12} , sc_{14} , sc_{15} , sc_{41} , sc_{42} , sc_{44} , sc_{75} 包含其它重复刀具，因此工步备刀前提下会发生死锁，根据定理 6 可知单步前瞻死锁检测即可保证系统处于安全状态。

表4.3 定理6示例

Tab.4.3 Example of theorem 6

机床	刀具需求顺序	重复刀具
M_1	$(T_1, T_2, T_4, T_1, T_5, T_4)$	T_1, T_4
M_2	$(T_4, T_3, T_5, T_1, T_6, T_4, T_2, T_1)$	T_1, T_4
M_3	(T_7, T_2, T_5, T_7)	T_7
M_4	$(T_4, T_1, T_4, T_1, T_4, T_1)$	T_1, T_4
M_5	$(T_1, T_8, T_7, T_1, T_6, T_3, T_7)$	T_1, T_7

基于上述定理，给出特殊刀具需求序列的死锁避免策略如表 4.4 所示。

表4.4 特殊刀具需求序列的死锁避免策略

Tab.4.4 Deadlock avoidance strategy of Special tool sequence

定理	刀具需求序列特征	死锁避免方策略
定理 4	每个刀具需求顺序中至多包含一把重复刀具	安全状态（工步备刀）
定理 5	重复刀具集中不包含重复刀具	安全状态（工步备刀）
定理 6	每个刀具需求顺序中至多包含两把重复刀具	死锁状态（工步备刀+单步前瞻）

4.4.3 无死锁调度算法

(1) 虚拟机床队列思想

工件流-刀具流并存的柔性制造系统中影响系统性能的因素较多，如某些工序存在可替代加工工序、工件的加工工序之间的加工顺序、某些刀具存在可替代加工刀具以及刀具分派的顺序等。这些因素给系统性能提高创造有利条件的同时，也为系统的调度控制增加了难度。由于工件在 FMS 中加工的过程可用排队理论和方法来描述，本文提出机床虚拟队列方法，与文献[135]中的工件虚拟队列方法相结合，大幅度降低工件流-刀具流综合调度问题的复杂性。

虚拟机床队列的基本思想是在虚拟加工工序队列基础上，考虑到刀具流柔性对系统性能的影响而提出的，其基本原理如图 4.13 所示。为了方便分析理解问题，现假设

所有刀具都按类存放于中央刀库的不同区域，由刀具机械手将各刀具运往目的机床。假设系统内每一类刀具前都存在一队列（集合），该队列中存放的是当前需求此类刀具的机床集合，称为虚拟机床队列，在调度开始时，机床前可加工工序集合为空，故各类刀具前的机床队列也为空，当工件流为各机床从虚拟工序队列中选定可加工工序后，假设选定工序加工所需的第一把刀具已备好，各机床开始加工，基于工步备刀前提，机床为下一工步准备刀具，按其准备情况将机床分为两类：一类是下一工步所需要刀具已经备好的机床集 1，一类是下一工步加工所需要刀具未准备好的机床集 2，对于前者，每当有刀具被运送到达目的机床时，确定各机床后续可加工工步所需要的刀具，并将此类机床加入到各类刀具（机床集 2 中各机床后续可加工工步所需要的刀具）前的机床队列中，然后等待刀具决策点到达。对于后者，将各机床加入到各类刀具（机床集 1 中各机床下一工步所需要的刀具）前的机床对列中，同时刀具决策点到达，也即各类刀具决策出其对虚拟机床队列中各刀具的分派顺序。

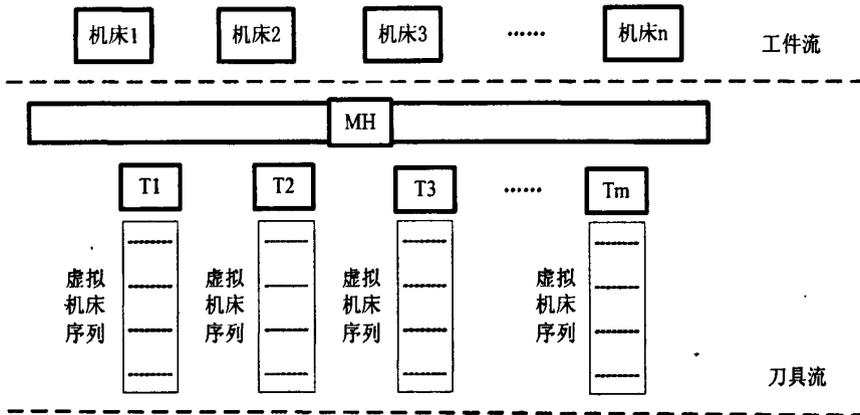


图 4.13 虚拟机床队列原理图

Fig4.13 Schematic diagram of virtual machine queue

从上述虚拟机床队列的思想不难看出，这种处理方法使得刀具失效、紧急工件进入、可替代刀具等重调度情况很容易处理。

(2) 无死锁调度算法流程

基于两层死锁避免策略和虚拟机床队列思想建立工步备刀下刀具流无死锁启发式调度算法如图 4.14 所示。在启发式调度中，调度决策点对机床及刀具的利用率及系统的加工性能有很大影响，决策点选取的不同，则调度结果不同，机床及刀具的利用率和系统的加工性能也就不同。对于一台机床或一把刀具，过早地决策其加工工件，将影响系统总体的协调，同样，过迟地决策，将影响机床和刀具的利用率并延误工件加工。所以合理地选取决策点，将提高系统的调度性能。

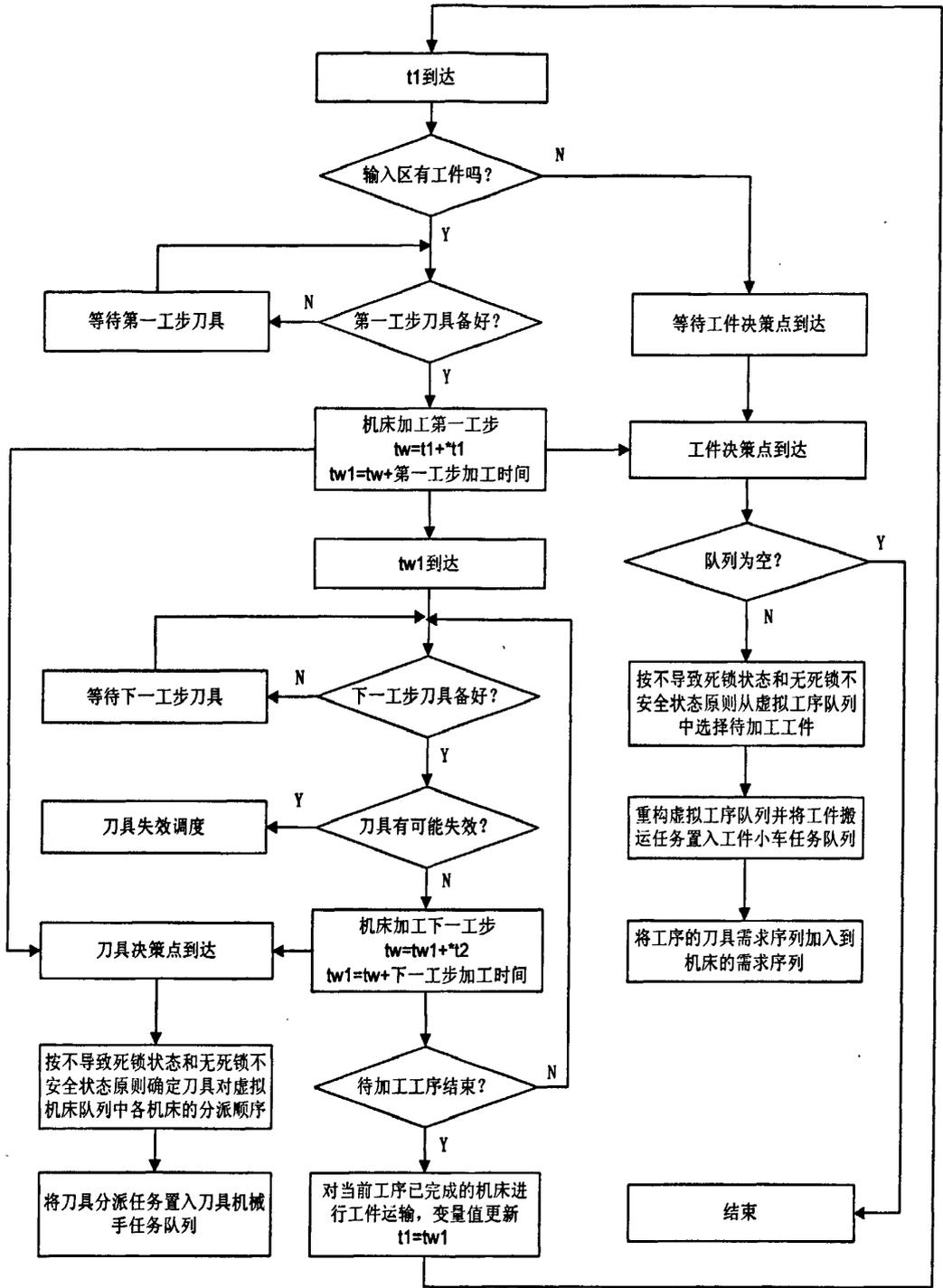


图 4.14 工步备刀下刀具流无死锁调度流程图

Fig4.14 Flow chart of deadlock-free scheduling in case of working step provisioning tools

工步备刀前提下，在各机床刀具需求序列中的第一把刀具准备好，机床就可以开始加工，后续工步不再使用的刀具即可释放。在此备刀前提下，系统的调度决策点由两部分组成：机床决策点和刀具决策点。对机床决策点，每当机床的输入缓冲区中工

件开始加工时，系统即按不导致死锁状态和无死锁不安全状态的原则从虚拟工序队列中为各机床决策下一待加工工序，然后将工件运输任务置入工件小车任务队列，等待将决策工件运往目的机床的输入缓冲区。对刀具决策点，每当机床上有工步开始加工时，则确定下一工步加工所需要的刀具，若刀具未准备好，则将此机床加入相应刀具的虚拟机床队列中，此类刀具按不导致死锁状态和无死锁不安全状态原则确定其对虚拟机床队列中各刀具的分派顺序，并将刀具运送任务置入刀具机械手的任务队列，等待运往目的机床。

4.4.4 刀具失效调度

在工步备刀情况下，刀具的磨损失效也即寿命换刀在备刀过程中已经考虑，故在此只对加工过程中的刀具破损失效给出处理策略，如图 4.15 所示。

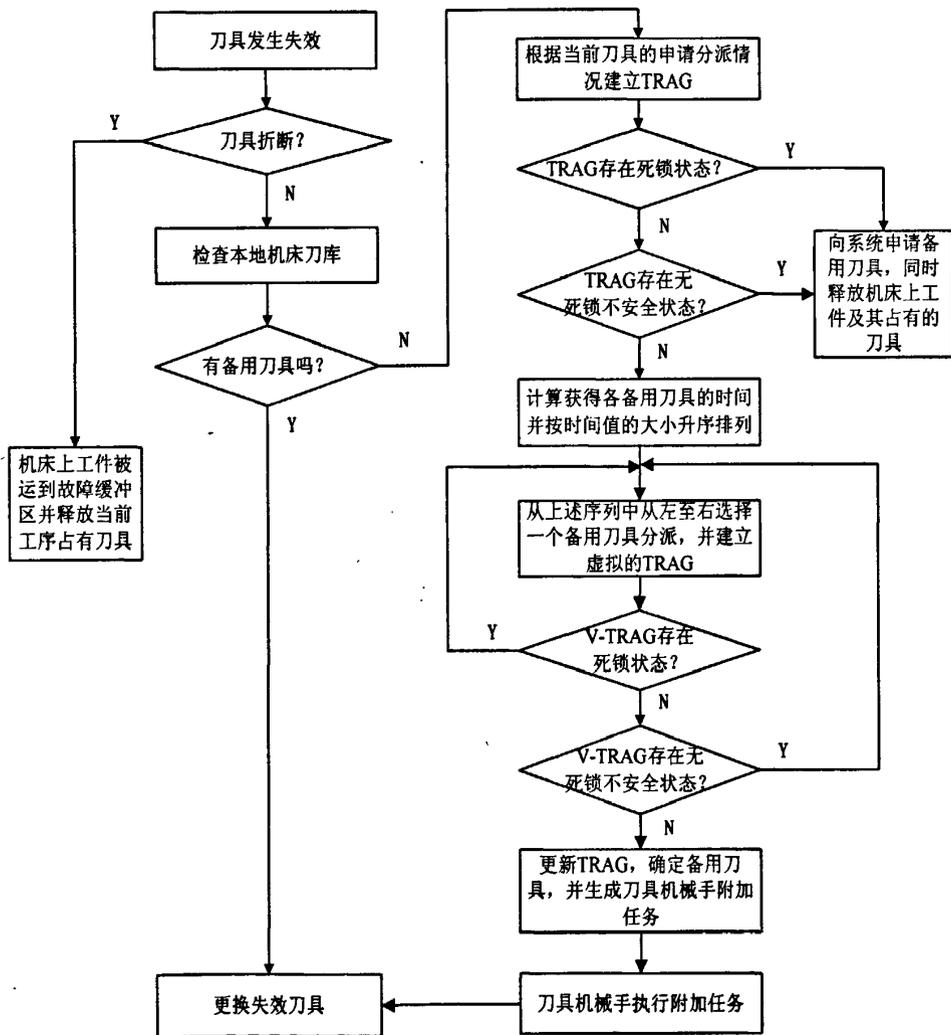


图 4.15 工步备刀下刀具流失效调度流程图

Fig.4.15 Scheduling flow chart with tools failure in case of working step provisioning tools

若刀具发生折断情况,此时需要把工件运走进行处理,并释放此加工工序所占用的刀具;若发生崩刃、碎断和裂纹等破损情况,则机床需向系统申请破损刀具的备用刀具,若本地机床刀库内有备用刀具,则由加工中心的自动换刀装置进行失效刀具更换,否则,系统根据当前刀具的申请分派状况,重新建立刀具资源分派图 TRAG,若 TRAG 中存在死锁状态或无死锁不安全状态,则说明此时系统中现有的备用刀具已经不存在可行的刀具分派方案来替换失效刀具,故需向系统申请备用刀具,并释放机床上的工件及其占有的刀具,待系统准备好备用刀具后,在将工件重新加入调度序列。若不存在死锁状态和无死锁不安全状态,说明当前系统的备用刀具存在可行的刀具分派方案,使得失效刀具得以替换,故此时计算获得备用刀具的时间,并按时间值的大小重新排列,为了减小刀具失效对系统全局的影响,首先选择获得时间最短的备用刀具,然后建立虚拟的 TRAG,对备用刀具分派之后的状态进行死锁状态检测和无死锁不安全状态检测,若不存在上述两种状态,说明此方案可行,也就确定了替换失效刀具的备用刀具,根据备用刀具当前的任务情况,为刀具机械手生成运送备用刀具任务,待刀具机械手执行完此任务后,失效刀具得以替换,从而以个别机床的短时间等待为代价来换取全局的协调。

4.5 实例分析

为验证无死锁不安全状态检测算法的有效性,以下给出了一个刀具分派示例。

1) 建立的 TRAG,如图 4.16(a)所示。

2) 确定刀具争用节点 T_2 的分派方案集,其中: $M_i(k)$ 表示机床 M_i 的第 k 个工步, $T_j^k(l_w)$ 表示 T_j 类刀具的第 k 个实例的寿命为 l_w ;

$$\text{方案 1: } \begin{cases} T_2^1(17): M_2(2) \rightarrow M_1(2) \rightarrow M_1(5) \\ T_2^2(5): M_2(4) \rightarrow M_2(6) \end{cases}$$

$$\text{方案 2: } \begin{cases} T_2^1(17): M_1(2) \rightarrow M_1(5) \rightarrow M_2(2) \\ T_2^2(5): M_2(4) \rightarrow M_2(6) \end{cases}$$

$$\text{方案 3: } \begin{cases} T_2^1(17): M_2(2) \rightarrow M_2(6) \rightarrow M_1(2) \rightarrow M_1(5) \\ T_2^2(5): M_2(4) \end{cases}$$

$$\text{方案 4: } \begin{cases} T_2^1(17): M_2(4) \rightarrow M_1(2) \rightarrow M_1(5) \\ T_2^2(5): M_2(2) \rightarrow M_2(6) \end{cases}$$

3) 建立虚拟的刀具资源分派图 V-TRAG;

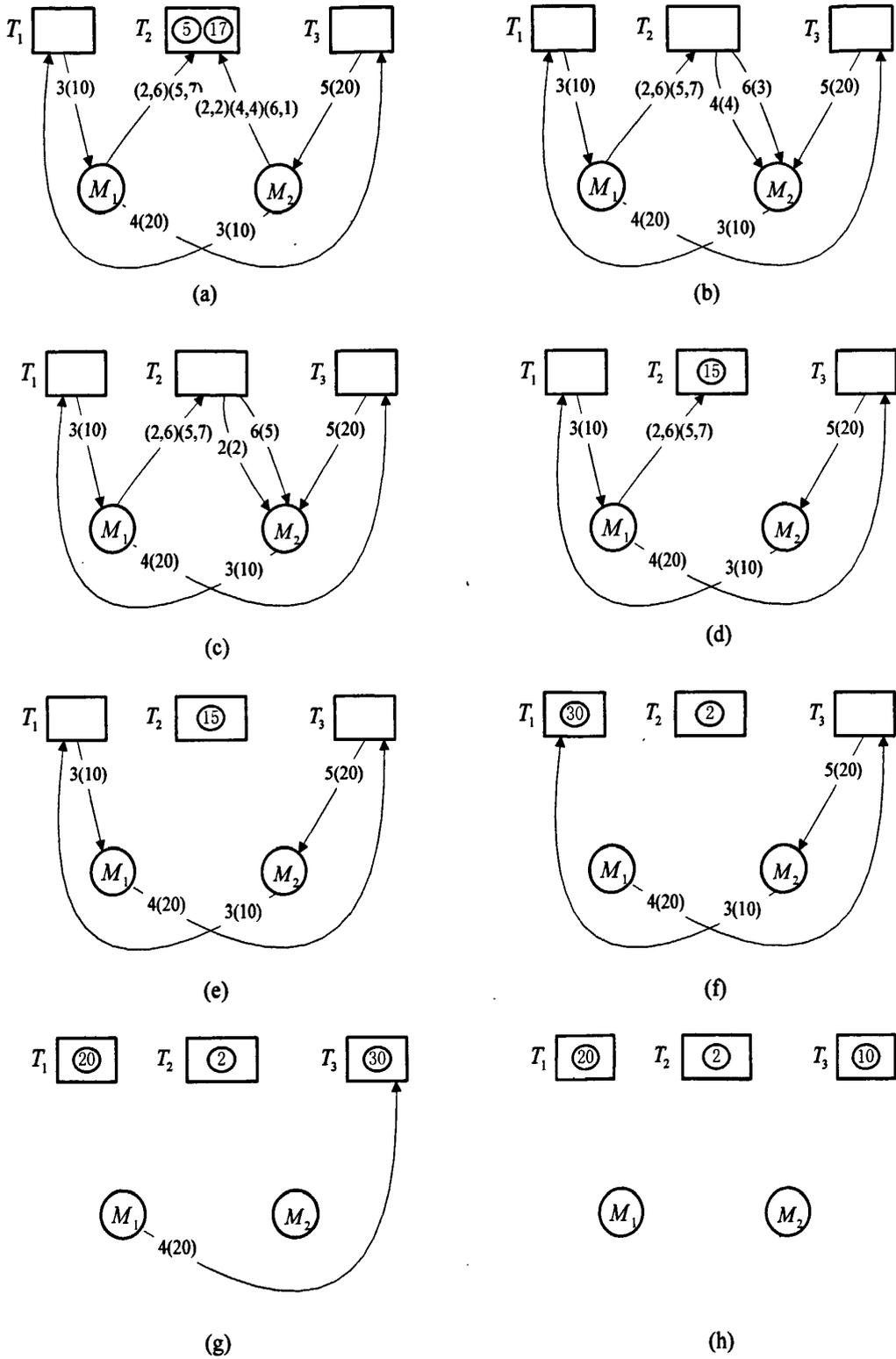


图 4.16 刀具分派过程示意图

Fig.4.16 Schematic diagram in the process of tools assignment

4) 从方案集中选择一种方案(如方案3)进行刀具分派,分派之后的 TRAG 如图 4.16(b)所示;

5) 图 4.16(b)为不可完全化简 TRAG,也即按选定的方案3进行刀具分派会导致系统陷入无死锁不安全状态,故选择其他分派方案(如方案1),分派之后如图 4.16(c)所示;

6) 对图 4.16(c)按简化规则1和4进行简化,简化后的 TRAG 如图 4.16(d)所示;

7) 对图 4.16(d)按简化规则2进行简化,简化后的 TRAG 如图 4.16(e)所示;

8) 对图 4.16(e)按简化规则1进行简化,简化后的 TRAG 如图 4.16(f)所示;

9) 对图 4.16(f)按简化规则3进行简化,简化后的 TRAG 如图 4.16(g)所示;

10) 对图 4.16(g)按简化规则6进行简化,简化后的 TRAG 如图 4.16(h)所示;

11) TRAG 可完全化简,说明存在一个刀具分派顺序使所有工件都顺利得到加工,故给定状态为安全状态。

4.6 本章小结

本章主要讨论了工步备刀前提下刀具流的无死锁调度问题,首先通过占有等待定义得出刀具流死锁只与机床 H 域和 W 域中的刀具有关,因而刀具流状态表现为系统中机床的 W 域和 H 域刀具的申请与占有状况,进一步将该状况用图论方法建立了刀具流状态演变模型,给出了刀具流死锁状态及无死锁不安全状态的定义。然后在刀具资源分派图 TRAG 基础上证明了刀具流死锁状态与 TRAG 中存在闭合强连通图之间的等价性,以及无死锁不安全状态与死锁机床之间的等价性,进而提出了刀具流死锁检测算法和无死锁不安全状态检测算法。在此基础上,给出了两层死锁避免方法,并针对特殊刀具需求序列的死锁避免进行了详细分析,实例表明该方法能够简单有效地避免死锁的产生。在刀具分派决策研究中提出了虚拟机床队列思想,并将其与死锁避免方法相结合,最终构建了工步备刀下的刀具流无死锁启发式调度算法,并对调度过程中的刀具失效给出了相应的处理策略,解决了工步备刀下的刀具流死锁避免和刀具实时分派问题。

第5章 工步备刀下的刀具流优化调度研究

启发式调度本质上是一种规则调度，因此所寻求到的调度解一般不可能是最优解或近优解。为此，本章在工步备刀前提下进一步研究刀具流的优化调度。以往的优化调度研究主要集中于工件流^[129-134]，也就是是工序的优化排序研究。在这类优化研究中，大多都不考虑刀具因素，而认为任意时刻刀具都是满足需要的。相对工件流的优化，刀具流的优化研究则不多见，这一方面的研究主要集中在刀具管理和刀具分派上，且在对刀具分派优化时，只局限于刀具流内部，不考虑工件流的影响。而在 FMS 运行中，刀具流并不是孤立存在的，它总是和工件流交织在一起，工件流驱动刀具流为之准备所需刀具，有限的刀具资源又制约着工件流的各项决策和运行过程，所以对刀具流优化调度的研究包括刀具实时分派的同时又不可避免地必须将工件的优化排序包括进来，也即刀具流调度优化的实质是工件流和刀具流的综合调度优化。

为此，本文构建了一个两层的优化策略来实现这一目标，首先通过对工件流和刀具流综合调度问题的描述，以系统完工时间最少为优化目标，建立工件流和刀具流综合调度数学模型。根据所分析问题的特殊性，提出了一类双重遗传算法，将工件流和刀具流的调度有机地统一到算法中。针对考虑刀具破损等故障情况的系统进程，进一步给出了一种基于双重遗传算法的准静态优化再调度策略。

5.1 综合调度问题的描述

工件流和刀具流综合调度控制的目标是如何实时地为所有工件安排合适的机床以及为其分派所需刀具，使得系统的总生产时间最短。为了既能够简化分析又能够反映问题实质，首先给出如下基本假设：

- 1) 每台机床同一时刻只能加工一个工序；
- 2) 工件的工步加工操作过程不能分解或中断；
- 3) 工件小车和刀具机械手不发生故障；
- 4) 每个操作的加工时间是确定的且事先已知。
- 5) 装夹定位工件等辅助时间包含在工件的加工时间里；
- 6) 工件的加工工艺路线是固定的

基于上述假设，建立工件流和刀具流综合调度问题的数学模型如下：

$$\text{Min}(C_{\max})$$

其中：

$$C_{\max} = \sum_i \sum_{d \in D_i} \sum_{j \in J_i} P_{ijm} Z_{idjm} + \sum_i \sum_{d \in D_i} \sum_{j \in J_i} \sum_{\lambda} a_{ij\lambda} \cdot (1 - \omega_{m\lambda}) \psi_{m\lambda}(Y) Z_{idjm} \quad (5-1)$$

subject to:

$$\sum_m Z_{idjm} = 1 \quad (5-2)$$

$$\sum_\lambda \omega_{m\lambda} \leq K_m \quad (5-3)$$

$$Z_{idjm}, \omega_{m\lambda} \in \{0, 1\} \quad (5-4)$$

式中, i —工件类别号; j —工序号; m —机床号; λ —刀具类别号; C_{\max} —系统生产时间; D_i — i 类工件集合; J_i — i 类工件工序集合; P_{ijm} — i 类工件的工序 j 在机床 m 上加工所需时间; Y —刀具集, 即 $\{y_\lambda\}$; $\psi_m(Y)$ — Y 刀具配置下机床 m 等待刀具 λ 的等待时间; $a_{ij\lambda}$ —值为 1 表示 i 类工件的工序 j 需用刀具 λ , 否则为 0; Z_{idjm} —值为 1 表示 i 类 d 工件的工序 j 被机床 m 加工, 否则为 0; ω_m —值为 1 表示机床 m 上有刀具 λ , 否则为 0; K_m —机床 m 的刀库容量。

模型中式 (5-1) 表示系统的生产时间, 包括机床正常不间断加工工件的时间与该机床在指定刀具配置下实时产生的刀具等待时间, 刀具流调度的目标就是尽量减小系统的生产时间; 式 (5-2) 表明每一工件的每一工序只能由一台机床完成, 而不能分解到不同机床上; 式 (5-3) 是指有限刀具库容量限制。

5.2 动态调度策略

本文提出了准静态再调度方法, 在此基础上进行动态调度过程, 具体描述为:

$$D_c = S_\beta^0 + \sum_{i=1}^n S_\beta^i$$

其中: D_c 为系统的动态调度过程;

S_β^0 为初始准静态调度;

S_β^i 为关键事件 i 发生后的准静态调度。

准静态再调度方法就是把一个动态的调度过程按关键事件分成多个调度区间, 每当有关键事件发生时, 根据当前的生产状态执行优化调度算法—准静态调度算法, 从而产生一个最佳的调度方案。它放弃了全局最优的概念, 通过准静态调度算法在线优化每个调度区间, 使系统在此区间内达到最优, 从而使调度方案适应复杂多变的动态环境。在此定义机床故障, 刀具破损, 机床恢复生产, 工件交货期改变, 紧急订单到达等为关键事件, 采用准静态再调度方法的动态调度流程如图 5.1 所示。

Step1: 根据生产任务, 启动准静态调度算法, 产生预调度方案并开始执行。

Step2: 判断系统是否出现关键事件, 如有, 则转入 *Step3*, 否则继续执行现有的调度方案。

Step3: 判断关键事件的影响, 决定是否继续执行预调度方案, 若刀具发生崩刃、碎断和裂纹等破损情况, 则采用相应的处理策略后, 可以继续执行预调度方案, 否则转入 Step4。

Step4: 修改再调度参数, 重新执行准静态调度算法, 产生再调度方案并开始执行调度方案。

Step5: 重复 Step2-Step4。

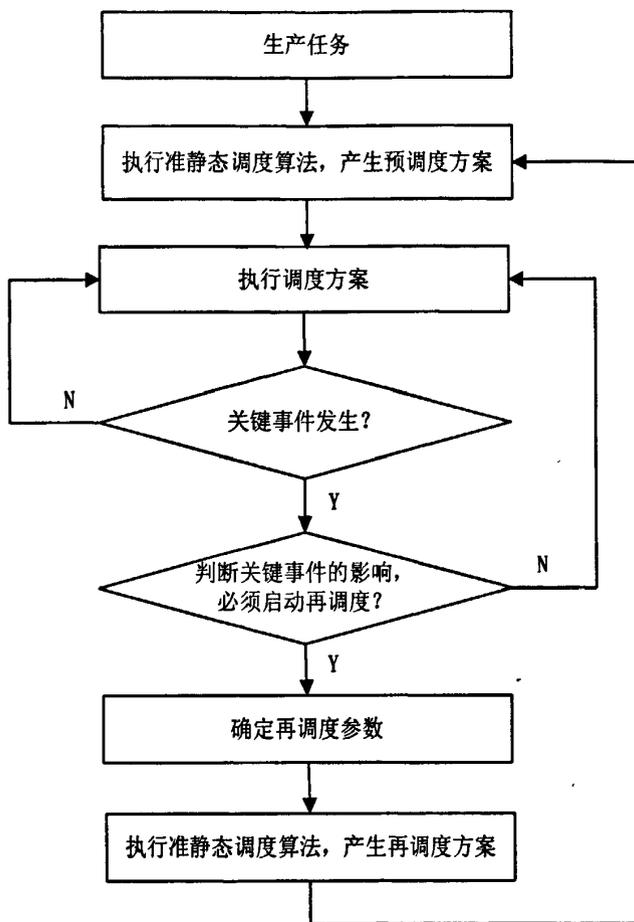


图 5.1 动态调度流程图

Fig.5.1 Flow diagram of dynamic scheduling

5.3 双重遗传优化算法的提出

5.3.1 综合调度思想

工件的加工工艺信息如图 5.2 所示, 其中工序层与机床层相对应, 表示加工工序所需要的机床资源, 工步层与刀具层相对应, 表示加工工步所需要的刀具资源, 如: 工件 P_1 的第一个工序 O_{11} 需要在机床 M_1 上加工, 其包含三个工步: O_{111} , O_{112} 和 O_{113} ,

其中第一个工步 O_{111} 需要 T_1 类或 T_9 类刀具加工, 第二个工步 O_{112} 需要 T_2 类或 T_8 类刀具加工, 第三个工步 O_{113} 需要 T_3 类刀具加工。

为确定一个可行的作业调度, 工件流首先为机床确定可行的不同的工序加工顺序, 然后驱动刀具流为之准备刀具, 对每个可行的工序加工顺序, 刀具流根据现有的刀具资源形成可行的不同的刀具分派方案, 不同的刀具分派又对应着系统不同的运行时间。可见一个作业调度的运行时间是由工件的加工顺序和刀具的分派顺序共同决定的, 此即为工件流和刀具流综合调度的基本思想, 它为下一步双重遗传算法的提出奠定基础。

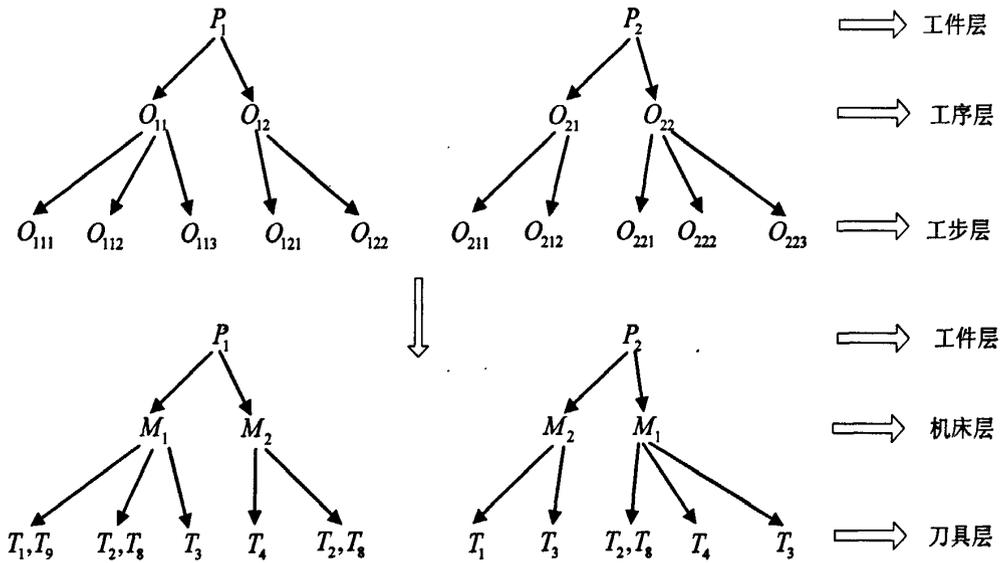


图 5.2 工件的工艺信息

Fig.5.2 Technological information of parts

5.3.2 双重遗传算法描述

双重遗传优化算法流程如图 5.3 所示, 外层优化即可行工序加工序列的最优化, 目标是使得系统运行时间最短。由于系统运行时间还要受到刀具分派的影响, 所以在外部优化中还不能够确定各种解的优劣性, 为此引入了内层优化, 即可行刀具分派方案的最优化, 其优化目标仍然是系统的运行时间最小化。由于同一个工序加工序列对应着多种不同的刀具分派途径, 不同的分派又对应着不同的系统运行时间, 所以这一层次的优化一方面可以作为外层优化的评判标准, 另一方面还可用于生成所需的刀具分派计划。当内层和外层优化同时满足终止条件时, 即可将全局优化结果输出, 用于指导系统的实际运行。输出的结果应当包括以机床甘特图描述的最优工序加工序列, 以及以表格形式描述的最优刀具分派方案。

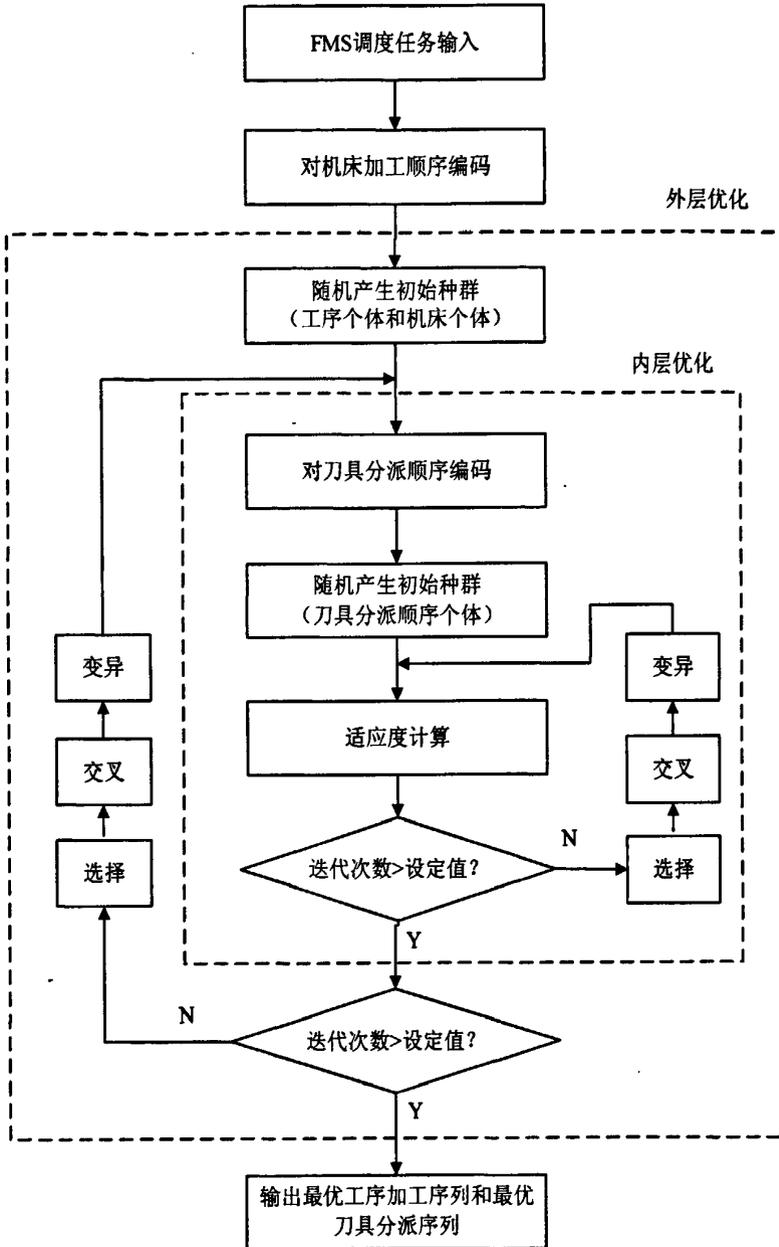


图 5.3 双重遗传优化算法

Fig.5.3 Double genetic optimization algorithm

5.4 双重遗传算子的设计

5.4.1 外层染色体编码

外层染色体编码也即机床加工顺序编码,在此设计两组染色体用来表示机床上工件的加工顺序,即:

$$CH_1 \text{ (工序号): } a_1, a_2, a_3, \dots, a_m$$

CH_2 (机床号): $b_1, b_2, b_3, \dots, b_m$

a_k 一为工序集 $O = \{O_{ij} | i=1, 2, \dots; j=1, 2, \dots\}$ 中的元素, 其中: $k=1, 2, \dots, m$, i 为工件号, j 为工序号。 b_k 一为机床集 $M = \{M_i | i=1, 2, \dots, n\}$ 中的元素, 且允许 $M_i = M_j (i \neq j)$ 。

例如: $CH_1: O_{11} O_{21} O_{22} O_{12}$

$CH_2: M_1 M_2 M_1 M_2$

则可知各机床上工序的加工顺序为:

$M_1: O_{11} O_{22}$

$M_2: O_{21} O_{12}$

因此 CH_1 和 CH_2 两条染色体能够同时反映工件的分组与排序情况, 然后根据内层染色体来计算系统的 Makespan 值。

5.4.2 外层适应度计算

(1) 内层染色体编码

内层染色体表示在当前外层染色体下各类刀具的分派顺序, 这些刀具分派顺序组合成系统的刀具分派方案, 故内层染色体编码即刀具分派方案编码。

设 $RQ_j(T_j)$ 为需求 T_j 类刀具的工步, N_j 为需求 T_j 类刀具的工步数之和, 则 $N_j = \sum_{\varphi \in \Phi} RQ_j(T_j)$, $\varphi = \{1, 2, 3, \dots\}$, $j = \{1, 2, 3, \dots, n\}$, 其中: φ 为工步编号集合。设 $ma = \max_{j \in \Phi} \{N_j\}$, 用列表示 $n \times ma$ 个工步位数, 用行表示每类刀具的实例, 则刀具分派顺序染色体可以表示为一个如图 5.4 所示形式的二维表, 这样就可以把 T_j 类刀具中各刀具实例的分派顺序作为一个任务单元放在表中, 表中未被工步占用的位置用 0 元素表示, 其中 c 表示每类刀具的刀具实例号。

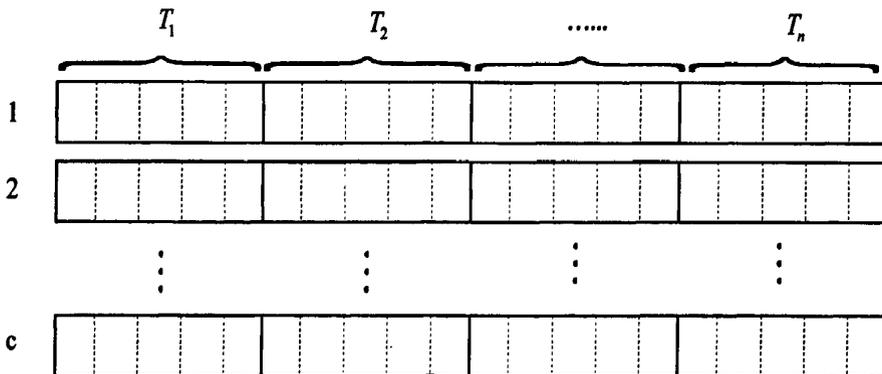


图 5.4 刀具分派顺序编码示意图

Fig.5.4 Coding schematic diagram of tool allocation sequence

(2) 初始种群的生成

设种群容量为 N ，即产生 N 个刀具分派方案染色体，其具体的生成算法流程如图 5.5 所示。

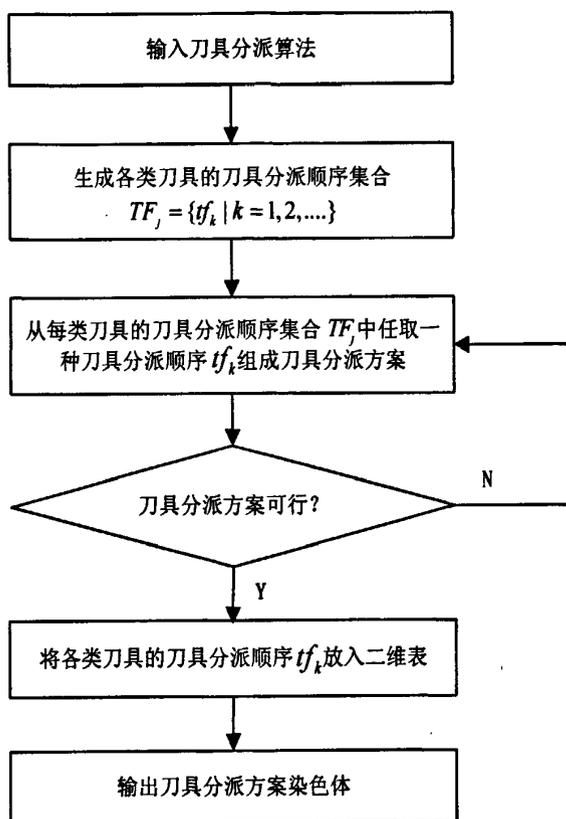


图 5.5 刀具分派方案染色体生成流程图

Fig.5.5 Flow chart of tool allocation scheme chromosomes generation

算法首先确定各类刀具的刀具分派顺序集合 TF_j ，也即在刀具寿命约束下，每类刀具都会有若干种刀具分派顺序，将每种分派顺序称为一个任务单元，从每个 TF_j 里任意取出一种任务单元即组成刀具分派方案，但是在生成各类刀具的刀具分派顺序集 TF_j 时只考虑了其刀具实例的可分派工步，并未考虑工件的工艺约束和机床加工顺序约束，故还须根据可行性进行判定方法进行刀具分派方案的可行性判定，若可行，则将各类刀具的任务单元放入图 5.4 所示的二维表中，形成一个刀具分派方案染色体，否则就要重新寻找分派方案。基于以上过程，下面给出刀具分派算法及刀具分派方案可行性的判定方法。

(3) 刀具寿命约束下的刀具分派算法

各类刀具 T_j ， $j=1,2,\dots,n$ 在其刀具寿命约束下的刀具分派算法如下：

Step1: 将 n 把 k 型刀具剩余寿命存入： $LT_{k1}, LT_{k2}, \dots, LT_{ki}, \dots, LT_{kn}$ ，其中： $i=1,2,\dots,n$ 为刀具编号， k 为刀具类型号， LT_{ki} 为 k 类型编号 i 刀具的剩余寿命；

Step2: 将需求 k 类型刀具的 m 个工步的加工时间存入: $LW_{k1}, LW_{k2}, \dots, LW_{ki}, \dots, LW_{km}$, 其中: $i=1, 2, \dots, m$ 为需求 k 类型刀具的工步编号, LW_{ki} 为编号 i 工步的加工时间;

Step3: 确定 m 个工步的编号组合, 存入数组 $S_j, j=1, 2, \dots, s$, 其中 $s = \sum C_m^i, i=1, 2, \dots, m$, 数组 S_j 中工步的加工时间之和存入 $LWS_j, j=1, 2, \dots, s$;

Step4: 将 n 把 k 型刀具的剩余寿命与 Step3 中得到的 s 个工步加工时间之和变量相减得到 $n \times s$ 个数据, 存入数组:

$$TK_i = LT_{ki} - LWS_j, i=1, 2, \dots, n, j=1, 2, \dots, s$$

设数组 TK_i 中的大于或等于 0 的元素所在的列数为 j , 则 k 类型编号 i 刀具可完成的工步组合信息为 S_j 。

Step5: 读出数组 TK_i 中满足 Step4 条件的 S_j , 将 S_j 数组中的非零元素与非 TK_i 中的 S_j 及非 S_j 数组中的各非零元素相减, 如出现零值, 即为有重复工步, 应避免进行刀具工步组合, 如果未出现零值, 且两相比较数组的非零元素之和等于或小于工步数之和, 则 Step6。

Step6: 若两相比较数组的非零元素之和等于工步数 m , 则为合格方案之一, 相应的刀具编号及工步编号存入数组 tf_k ; 如果两相比较数组的非零元素之和小于工步数 m , 保存相应的刀具编号及工步编号, 以供下一步选择。

Step7: 数组 tf_k 中存储的即为各类刀具实例的分派顺序。

(4) 刀具分派方案的可行性判定

刀具的分派必须符合机床加工顺序约束和工艺约束, 因此需要对遗传操作中生成的刀具分派方案的可行性进行判定。建立可行性判定方法如下:

首先根据综合调度的基本约束条件, 即工艺约束, 机床加工顺序约束以及刀具分派顺序约束得到如图 5.6 所示的加工顺序约束关系图, 其中 T_j^k 表示 T_j 类刀具的第 k 个刀具实例。

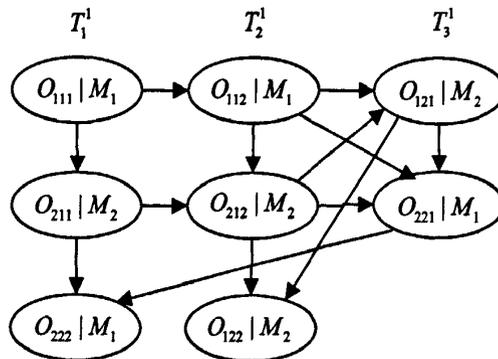


图 5.6 加工顺序约束图

Fig.5.6 Constraints to machining sequence

图中的节点代表工步及其加工机床，箭头方向表示加工顺序，当一个节点有多个输入节点时，只有当所有的输入节点均加工完毕并释放占有的加工资源（机床和刀具）后，该节点才能获得加工资源进行加工。

由上述过程可以看出，当图论中的所有节点均为可加工节点时，刀具分派顺序才是可行的。将图论中节点间的连接关系用工步优先关系矩阵表示如图 5.7 所示。

$$\begin{matrix}
 & O_{111} & O_{112} & O_{121} & O_{122} & O_{211} & O_{212} & O_{221} & O_{222} \\
 \begin{matrix} O_{111} \\ O_{112} \\ O_{121} \\ O_{122} \\ O_{211} \\ O_{212} \\ O_{221} \\ O_{222} \end{matrix} & \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}
 \end{matrix}$$

图 5.7 工步优先关系矩阵

Fig.5.7 Precedence relation matrix of working step

根据图论思想，初始可加工节点加工完后，下一步可加工节点可通过下式求得：

$$[\pi_1 \pi_2 \pi_3 \cdots \pi_n] \& [R - I] = [\rho_1 \rho_2 \rho_3 \cdots \rho_n]$$

其中：I 为单位矩阵，R 为工步优先关系矩阵，R 中的元素为 r_{ki} 。

$$\rho_i = (\pi_1 \wedge r_{1i}) \wedge (\pi_2 \wedge r_{2i}) \wedge \cdots \wedge (\pi_n \wedge r_{ni}), \quad i = 1, 2, \dots, n$$

式中：

$$\pi_k \wedge r_{ki} = \begin{cases} 1, & \text{如果 } r_{ki} = 0 \text{ 或 } r_{ki} = 1, \pi_k = 1, \quad k = 1, 2, \dots, n. \\ 0, & \text{其它} \end{cases}$$

当 $r_{ki} = 0$ 时，说明节点 π_k 不是节点 r_i 的输入节点，此时 r_i 节点的可加工性与节点 π_k 无关，因此定义 $\pi_k \wedge r_{ki} = 1$ 。

如果系统经过 n 次计算后，仍有节点为 0，说明约束图中有不可加工节点存在，故刀具分派顺序是不可行的，否则刀具分派方案可行。

$[\pi_1 \pi_2 \pi_3 \cdots \pi_n]$ 为各加工节点，当 π_k 为初始可加工节点时， $\pi_k = 1$ ，否则为 0，例如： $[1000010]$ 说明第 1 个和第 7 个为初始可加工节点。

(5) 内层适应度计算

在双重遗传算法求解生产调度问题中，为了评价刀具分派方案的优劣，需要计算每一个可行刀具分派染色体的 Makespan 值，即

$$C_{\max} = \max \{C_1, C_2, \dots, C_i, \dots, C_n\}$$

式中 C_{\max} 为系统的 Makespan 值, C_i 为第 i 台机床完成其上的所有工件需要的时间。其中 C_i 的计算应满足工艺约束, 机床加工顺序约束以及刀具分派顺序约束, 也即图论中每一个节点的起始加工时间应取所有输入节点中最晚释放加工资源 (机床和刀具) 的时间作为该节点的开工时间。

设: t_{ps} 为当前工步的开工时间;

t_{amps} 为同一机床上前一工步的结束时间;

t_{atps} 为当前工步所使用的刀具的释放时间;

t_{awps} 为同一工序前一道工步的结束时间;

如果当前工步为第一工步, 则当前工步的开工时间为: $t_{ps} = \max \{t_{amps}, t_{atps}\}$ 否则工步的开工时间为: $t_{ps} = \max \{t_{amps}, t_{awps}, t_{atps}\}$ 。

(6) 遗传操作

选择算子: 采用两两复制和最佳个体相结合的方法。

交叉算子: 交叉表现了利用优良染色体进行杂交产生新个体的随机过程, 以染色体的一个任务单元为单位, 这样可以将若干刀具分派方案中某类刀具的任务单元互换, 避免刀具分派顺序的冲突。采用均匀交叉方式, 设两个父本分别为:

$$CH_3(1) = (T_1(1), T_2(1), \dots, T_i(1), \dots, T_n(1))$$

$$CH_3(2) = (T_1(2), T_2(2), \dots, T_i(2), \dots, T_n(2))$$

其中: $T_i(\beta), i=1, 2, \dots, n, \beta=1, 2$ 表示 β 父本中 i 类刀具的任务单元。

随机生成交叉位置 $\text{Pos} = [\delta_1, \delta_2, \dots, \delta_n]$, 其中 δ_i 为整数 0 或 1, 交叉后产生的新个体为:

$$CH_3(3) = (T_1(3), T_2(3), \dots, T_i(3), \dots, T_n(3))$$

$$CH_3(4) = (T_1(4), T_2(4), \dots, T_i(4), \dots, T_n(4))$$

式中:

$$T_i(4) = T_i(1), T_i(3) = T_i(2), \quad i > 0 \text{ 且 } i = i * \delta_i。$$

变异算子: 变异虽然以很小的概率发生, 但是它对防止搜索得到的解陷入局部次优解, 以及抑制遗传算法早熟现象的发生有极其重要的作用。刀具分派问题的变异算子主要是各类刀具的刀具实例在分派顺序上的变异, 让已经安排的刀具分派顺序随机变成另一分派顺序。在这里把染色体的任意一个任务单元中的任意一把刀具实例的分派顺序与另一个随机选取的可行的刀具分派顺序进行互换操作。

5.4.3 外层遗传操作

选择算子: 采用两两复制和最佳个体相结合的方法。

交叉算子: 对染色体 CH_2 可采用通常的交叉运算方式, 即首先随机地从种群中取

出两个染色体 CH_{2i} 和 CH_{2j} ，再随机生成一个交叉位置，将交叉位置左边或右边的基因互相交换位置，从而得到一对新的染色体。例如：

$$CH_{2i}: M_1 M_3 M_2 | M_1 M_2$$

$$CH_{2j}: M_3 M_2 M_1 | M_3 M_1$$

将 CH_{2i} 和 CH_{2j} 的交叉位置“|”右侧的基因互换位置，得到的新染色体如下：

$$CH_{2i}: M_1 M_3 M_2 M_3 M_1$$

$$CH_{2j}: M_3 M_2 M_1 M_1 M_2$$

而染色体 C_i 属于随机排序问题，其交叉算法需进行修正，具体过程参照文献[129]。

变异算子：对于染色体 CH_1 ，由于各工序在染色体中仅允许出现一次，其基因无法变异。变异算法应用于染色体 CH_2 ，可将其机床号用可替代机床号替换。

5.5 实例分析

下面以实例来验证双重遗传优化算法的有效性。本实例中，柔性制造系统的组成主要包括：三台工件小车，一个刀具机械手，五台机床，编号分别为 M_1, M_2, M_3, M_4, M_5 。系统同时加工 4 类工件，需要 6 类刀具，其中刀具数量配置有三种方案。

方案一：每类刀具配置 1 把；

方案二：1-5 类刀具每类均配置两把，第 6 类刀具配置 1 把；

方案三：每类刀具均配置 3 把。

系统的工艺参数如表 5.1 所示，双重遗传算法的基本参数如表 5.2 所示。

表5.1 工艺数据

Tab.5.1 Procedure parameters

工件类型	工序序号	加工机床号	工步序号(刀具类型号/刀具使用时间 min)	
			1	2
1	1	1	1/20	2/18
	2	3	3/29	2/32
	3	5	4/21	5/45
2	1	2	6/51	2/37
	2	4	3/25	5/25
	3	3	4/39	1/18
3	1	4	1/42	4/31
	2	1	5/29	3/33
4	1	5	3/20	5/24
	2	2	2/25	6/27

表5.2 遗传算法的基本参数

Tab.5.2 Basic information of gegenitic algorithm

外层参数	N1=10	Pc1=0.8	Pm1=0.01	Counter=30
内层参数	N2=10	Pc2=0.8	Pm2=0.01	Counter=30

表 5.3 不同刀具配备下的性能指标

Tab.5.3 Performance indices under different tools configurations

刀具数量配备 方式	机床利用率					系统的完工 时间 (min)
	M_1	M_2	M_3	M_4	M_5	
方案一	72.09%	82.84%	60.05%	73.21%	57.74%	383
方案二	81.85%	94.59%	76.13%	92.83%	71.66%	310
方案三	81.85%	94.59%	76.13%	92.83%	71.66%	310

表 5.4 方案二下不同方法的总体指标

Tab.5.4 Results obtained with different methods for scheme 2

算法	换刀次数	等刀时间(min)	完工时间(min)
双重遗传算法	19	128	310min
文献[82]中算法	15	149	366min

表 5.5 刀具分派方案

Tab.5.5 Tool assignment scheme

刀具编号	刀具分派顺序	刀具编号	刀具分派顺序
1	$O_{211} \rightarrow O_{111} \rightarrow O_{432} \rightarrow O_{532}$	7	$O_{612} \rightarrow O_{512} \rightarrow O_{231} \rightarrow O_{131}$
2	$O_{611} \rightarrow O_{511}$	8	$O_{431} \rightarrow O_{331}$
3	$O_{412} \rightarrow O_{222} \rightarrow O_{312} \rightarrow O_{721} \rightarrow O_{821}$	9	$O_{322} \rightarrow O_{132}$
4	$O_{212} \rightarrow O_{112} \rightarrow O_{122}$	10	$O_{712} \rightarrow O_{812} \rightarrow O_{621} \rightarrow O_{422} \rightarrow O_{232} \rightarrow O_{521}$
5	$O_{811} \rightarrow O_{221} \rightarrow O_{421} \rightarrow O_{321} \rightarrow O_{522}$	11	$O_{411} \rightarrow O_{511} \rightarrow O_{722} \rightarrow O_{822}$
6	$O_{711} \rightarrow O_{121} \rightarrow O_{622}$		

表 5.3 给出了不同的刀具配备下系统的完工时间及加工设备的利用率（不考虑机械手换刀等辅助时间），不难看出，当系统中的刀具资源较为紧张（方案一）时，系统总加工时间达到 383min，各机床利用率较低；而当采用十分宽松的刀具配置（方案三）时，优化解的总加工时间有了显著减少，只有 310 min，降幅达 19%，且各机床利用率也得到了大幅提高，不过这种效率的提高是以刀具资源的高冗余为代价的；将刀具资源较为紧张的方案二与方案三对应的优化解对比可见，二者的各项参数完全一致，但刀具数量却减少了 38.8%，也即在刀具资源较少的方案二中所提出的双重遗传算法仍然能找到最优解。

表 5.4 给出了方案二下不同方法的总体指标（不考虑机械手换刀等辅助时间），可见，与传统的规则调度方法相比，采用本算法可使系统总的完工时间减少 15.3%，等刀时间下降 14.1%，而换刀次数略有增加。

表 5.5 和图 5.8 详细给出了方案二的优化解对应的刀具分派方案顺序及工件的加工顺序甘特图。

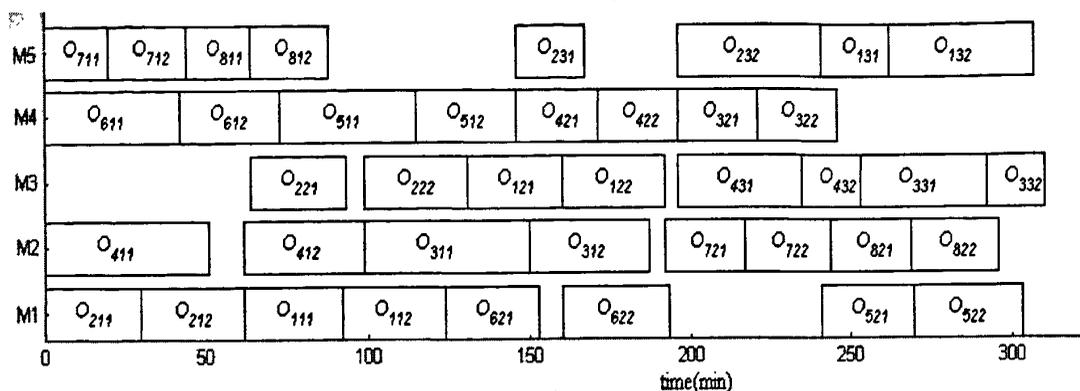


图 5.8 工件的加工顺序

Fig.5.8 Machining sequence of parts

5.6 本章小结

为解决柔性制造系统中工件流与刀具流并存情况下的调度优化问题，以系统完成时间最短为优化目标，建立了工件流-刀具流综合调度数学模型，提出了一种双重遗传算法对模型进行优化求解。外层遗传优化负责可行工序加工序列的求解，内层遗传优化进行最优可行刀具分派方案的搜索，搜索结果的适应度则作为外层优化的评判标准。实例分析结果表明：双重遗传算法在取得各工件优化排序的同时，还获取了各类刀具的优化分派，与传统的规则调度相比，系统的完工时间及等刀时间分别减少了 15.3% 和 14.1%。

第6章 仿真运行结果分析与评价

本章主要对工序备刀情况的启发式调度算法以及工步备刀情况的启发式调度算法及优化调度算法在所设计的仿真实例条件下进行仿真试验,并对其运行结果进行分析,仿真的目的是评价各种启发式调度规则的调度性能,并验证基于工序备刀和基于工步备刀的启发式调度算法及优化调度算法的有效性,最后对各启发式调度算法和优化调度算法做出综合评价。

6.1 系统仿真运行环境

6.1.1 系统的性能指标

刀具流调度的目标就是如何实时地为所有工件安排合适的机床以及为其分派所需刀具,从而取得较优的性能。一方面使系统在有限资源(机床,刀具,工件小车,刀具机械手)约束下能及时完成给定的加工任务,尽量满足加工任务中工件的交货期要求,另一方面尽可能减少生产时间,使系统具有较高的生产率。因此本文主要用以下几个指标来描述系统的性能:

1) 工件的平均流通时间 (Mean Flow Time, MFT)

工件的平均流通时间是指工件在系统中平均花费的时间,它反映了工件在系统中的驻留时间,能较全面的反映系统的生产时间,其数学表达如下:

$$MFT = (1/n) \sum F_i$$

其中: $F_i = C_i - A_i$ 为工件 i 的流通时间, C_i 为工件 i 的完成时间, A_i 为工件 i 的到达时间, n 为工件的个数。

2) 工件的平均延误时间 (Mean Tardiness Time, MTT)

工件的平均延误时间是指工件在系统中的完成时间与其交货期的差值,它反映了工件是否满足交货期的要求,其数学表达如下:

$$MT = (1/n) \sum L_i$$

其中: $L_i = \max\{0, (C_i - D_i)\}$ 为工件 i 的延误时间, C_i 为工件 i 的完成时间, D_i 为工件 i 的交货时间, n 为工件的个数。

3) 平均等刀时间 (Mean Waiting Time for Tool, MWTT)

平均等刀时间是指系统在加工任务过程中,机床平均等待刀具的时间,其值的大小与刀具配置关系密切,是系统获得较优的刀具配置方案的基础,其数学表达式如下:

$$MWTT = (1/n) \sum WT_i$$

其中: WT_i 为机床 M_i 在指定刀具配置下实时产生的刀具等待时间, n 为机床的台数。

4) 平均刀具交换次数 (Mean Tool Exchange Frequency, MTEF)

平均刀具交换次数是指在任务加工过程中, 如果所需要的刀具不在本地机床刀库, 需要向中央刀库或其他机床刀库平均借用刀具的次数, 其值的大小反映了系统在加工过程刀具的运输频率, 其数学表达式如下:

$$MTEF = (1/n) \sum W_i$$

其中: W_i 为机床 M_i 的刀具交换次数, n 为机床的台数。

5) 平均机床利用率 (Average Machine Utilization, AMU)

平均机床利用率主要用于测量在任务加工过程中各机床的平均利用情况, 考虑各机床的负荷是否平衡, 其数学表达式如下:

$$AMU = (1/n) \sum ST_i$$

其中: $ST_i = CM_i / CM$ 为机床 M_i 的利用率, CM_i 为机床 M_i 的累积运行时间, CM 为 FMS 的总生产时间, n 为机床的台数。

为了既能够简化分析又能够反映问题实质, 针对刀具流调度提出 FMS 的运行假设如下:

- 1) 零时刻所有工件处于系统外等待加工;
- 2) 一个工件不能同时在两台机床上加工, 且一台机床不能同时加工两个工件;
- 3) 工件小车每次只运输一个工件;
- 4) 运刀机械手每次只运输一把刀具;
- 5) 工件的加工工艺路线是固定的;
- 6) 装夹定位工件等辅助时间包含在工件的加工时间里;
- 7) 工件小车和刀具机械手不发生故障;
- 8) AGV 等待申请时有两类: 一类是将工件运进机床的输入缓冲区, 一类是将工件从机床的输出缓冲区运走, 这时为了防止机床发生阻塞, 后者具有优先权;
- 9) 工件小车和刀具机械手完成任务后, 停在附近机床或装卸站。

6.1.2 仿真运行案例

本实例中, 柔性制造系统的组成如图 6.1 所示。系统主要包括:

- (1) 机床 5 台, 每台机床都有其机床刀库;
- (2) 三台工件运输小车, 一个缓冲站及一个刀具进出站;
- (3) 一个中央刀库及一把刀具机械手。

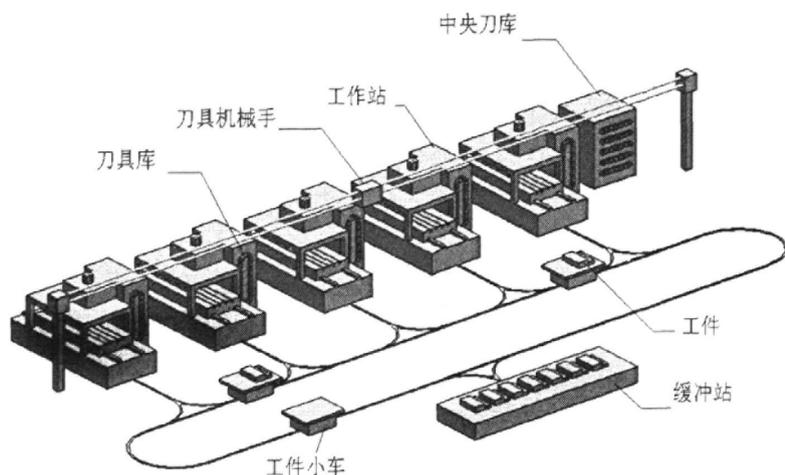


图 6.1 柔性制造系统布局

Fig.6.1 Layout of flexible manufacturing system

系统加工 5 种类型共 64 个工件,其中一类工件 16 个,二类工件 14 个,三类工件 10 个,四类工件 12 个,五类工件 12 个。工件的具体工艺信息如表 6.1 所示。

表6.1 工艺数据

Tab.6.1 Procedure parameters

工艺信息	生成方法
每类工序包含的工序数量	服从[2, 4]的均匀离散分布
每个工序的加工机床	服从[1, 4]的均匀离散分布
每个工序包含的工步数量	服从[2, 5]的均匀离散分布
每个工步的加工时间	服从[6, 30]的均匀离散分布
每个工步使用的刀具类型号	服从[1, 15]的均匀离散分布

每类型工件的加工顺序是给定的,每类工件中包含工序的数量服从[2, 4]的均匀离散分布,每个工序的加工机床服从[1, 4]的均匀离散分布,但同一工件的相邻两道工序的加工机床不允许重复,每个工序包含的工步数量服从[2, 5]的均匀离散分布,每个工步的加工时间服从[6, 30]的均匀离散分布,每个工步使用的刀具类型号从[1, 15]的均匀离散分布中产生,但同一工序的相邻两个工步所使用的刀具类型号不允许重复。工件的交货期= K *加工时间,其中: $K=2$ 。

系统配有 15 种刀具,每种 2 把,编号分别为 1-30,每把刀具的寿命为 200 分钟,每次加工之前需检查剩余刀具寿命是否足够完成接下来的加工。设机床刀库的容量为 7 把,各机床刀具库的初始配置如下:(数字为刀具编号)。

机床 1 刀库---1, 3, 5, 8, 10, 16

机床 2 刀库---4, 6, 7, 12, 14, 17

机床 3 刀库——2, 15, 21, 23, 25, 28, 30

机床 4 刀库——9, 11, 13, 18, 20, 24

机床 5 刀库——19, 22, 26, 27, 29

6.2 工序备刀下启发式调度运行结果及分析

本节主要针对系统运行中无故障和有故障两种情况,对 6 个工件载入规则和 3 个刀具需求选择规则生成的 18 个试验情形进行仿真运行并进行统计分析,以验证工序备刀下启发式调度算法的有效性,及评价各启发式调度规则的调度性能。其中,18 个试验情形,每一种情形重复试验 10 次,系统的各性能指标取 10 次试验的平均值。首先简要介绍一下各启发式调度规则。

1) 工件载入规则

交货期最早 EDD (Earliest Due Date): 该规则确定交货日期最早的工件首先接受服务,以使得该工件能尽早完成整个生产过程。

工序加工时间最短 SPT (Shortest Processing Time): 该规则使得机床在申请加工的工序队列里选择工序加工时间最短的工件进行服务。

先申请先服务 FCFS (First Come First Serve): 该规则规定先进入虚拟工序队列的工件先接受机床服务。

刀具交换次数最少 LEFT (Least Exchange Frequency of Tools): 该规则优先选择向其它机床申请刀具数量最少的工序作为本机床的下一道待加工工序,以减少刀具交换次数,减小刀具流负荷,进而改善系统性能。

剩余工序数最少 FNOP (Fewest Number of Operations): 该规则选择剩余工序数最少的工件首先接受服务,以便该工件尽早完成加工过程,使系统的在制品数减少。

松弛量最小 SLACK (Least Amount of Slack): 该规则选择松弛量最小的工件首先接受服务,工件松弛量=交付期-当前时刻-剩余加工时间。

2) 刀具需求选择规则

交货期最早 EDD (Earliest Due Date): 该规则规定刀具首先分派给交货日期最早的工件所在的机床,以使得该工件能尽早完成整个生产过程。

工序加工时间最短 SPT (Shortest Processing Time): 该规则规定刀具首先分派给工序加工时间最短的工件所在的机床。

目标机床当前工序加工完成时间最早 EFCO (Earliest Finish for Current Operation): 该规则规定刀具优先分派给机床队列(任务)中当前工序加工完成时间最早的机床,以减少机床等待刀具的时间,提高系统的生产效率。

6.2.1 无故障的启发式调度运行结果

假设系统在运行过程中无机床故障和刀具破损发生,依据不同的工件载入规则

和刀具需求选择规则得到的系统的各性能指标如图 6.2-6.6 所示。

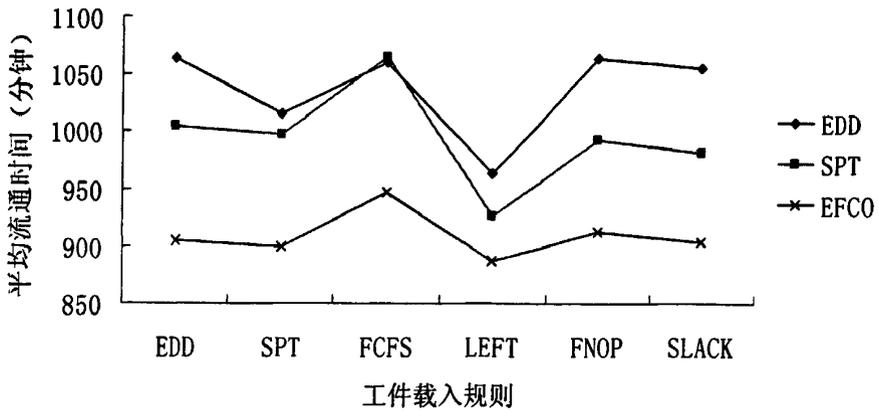


图 6.2 工序备刀下无故障的平均流通时间

Fig.6.2 MFT without failure in case of procedure provisioning tools

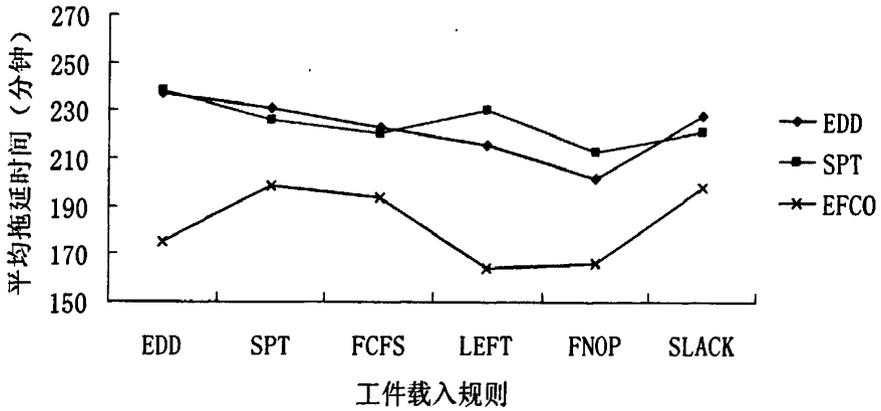


图 6.3 工序备刀下无故障的平均拖延时间

Fig.6.3 MTT without failure in case of procedure provisioning tools

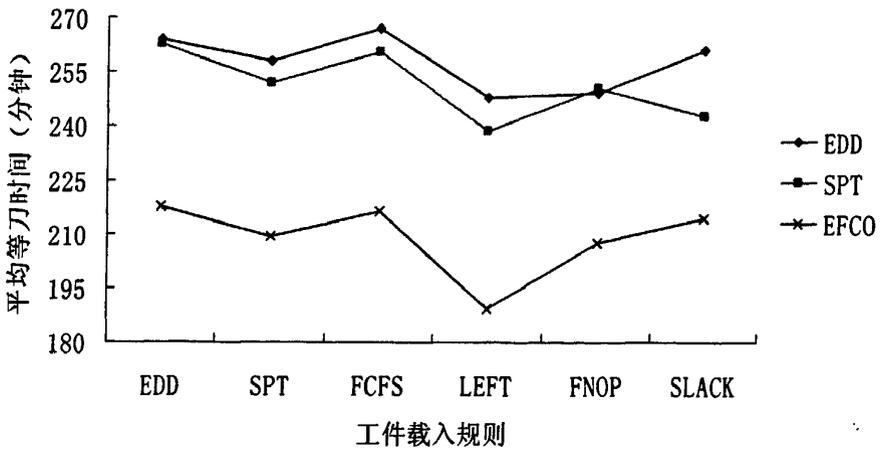


图 6.4 工序备刀下无故障的平均等刀时间

Fig.6.4 MWTT without failure in case of procedure provisioning tools

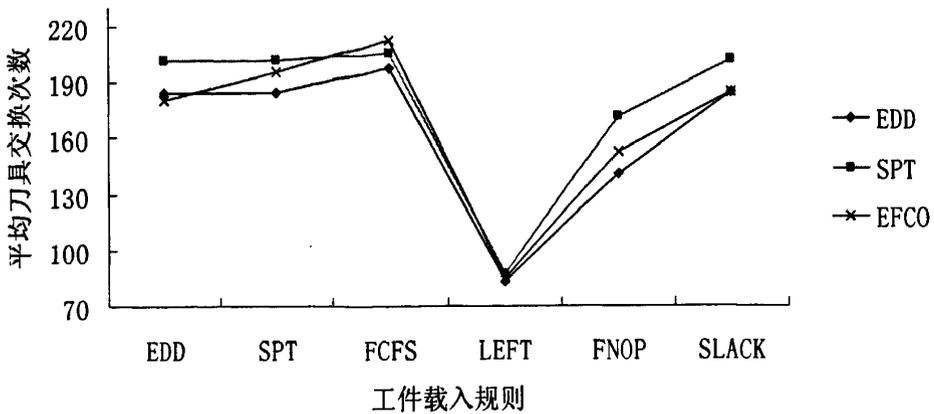


图 6.5 工序备刀下无故障的平均刀具交换次数

Fig.6.5 MTEF without failure in case of procedure provisioning tools

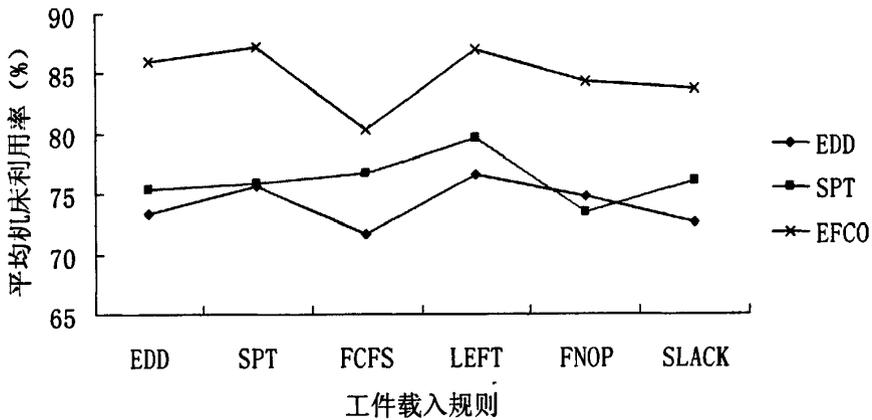


图 6.6 工序备刀下无故障的平均机床利用率

Fig.6.6 AMU without failure in case of procedure provisioning tools

为了分析无故障情况下不同的工件载入规则PLR和刀具需求选择规则TRSR对工序备刀下的系统各性能指标的影响,采用双因素方差分析法进行方差分析,结果如表6.2所示。其中:第一列的p值表示工件载入规则对系统性能的影响,第二列的p值表示刀具需求选择规则对系统性能的影响,第三列的p值表示两个因素PLR和TRSR共同作用对系统性能的影响。

从表6.2可知,工件载入规则和刀具需求选择规则对系统的平均流通时间,平均拖延时间和平均机床利用率都有影响,但相对于刀具需求选择规则,工件载入规则的影响小一些,这主要是由于在工序备刀前提下,工件只有在工序所需要的刀具都备齐后才可以开始加工,加工完毕后所占有的刀具才能释放,这致使刀具资源的利用率大大降低,而刀具的可得性是由刀具需求选择规则所决定的,故刀具需求选择规则对系统平均流通时间,平均拖延时间和平均机床利用率的影响更大。对于系统的等刀时间,

则受工件载入规则和刀具需求选择规则共同的影响。对系统的平均刀具交换次数，工件载入规则和刀具需求选择规则对其都有影响，但相比于工件载入规则，刀具需求选择规则的影响可以忽略不计。这说明若在机床选工序时，充分考虑所选工序对后续刀具流的影响，可以很大程度上改善刀具流的负荷，改善系统的性能。

表 6.2 工序备刀下无故障 PLR-TRSR 方差分析结果

Tab.6.2 Summary of ANOVA results without failure in case of procedure provisioning tools

系统性能指标	工序备刀下的 p 值		
	PLR	TRSR	Interaction
MFT	0.0412	0.0025	0.7930
MTT	0.0407	0.0028	0.8317
MWTT	0.0352	0.0051	0.0366
MTEF	0.0034	0.0431	0.6154
AMU	0.0463	0.0039	0.9025

由图 6.2-6.3 和图 6.6 可知:在几类刀具需求选择规则中, EFCO 规则与大部分工件载入规则组合都可以得到较优的平均流通时间、平均拖延时间和平均机床利用率, 这主要是由于 EFCO 规则在决策刀具的分派时考虑了对工件流效率的影响, 使得系统中最迫切需要刀具的机床能迅速得到刀具, 从而其待加工工序能尽早进入加工状态, 减少了机床等待刀具的时间, 提高了系统的生产率, 也即提高了系统的平均机床利用率, 从而验证了本文提出的 EFCO 规则的有效性。虽然工件载入规则对这几个性能指标的影响相对小些, 但是从图 6.2-6.3 和 6.6 仍可以看出 LEFT 作为工件载入规则与大部分刀具需求选择规则组合都可以得到较优的平均流通时间、平均拖延时间和平均机床利用率。

由图 6.4 可知, 系统的平均等刀时间在刀具需求选择规则 EFCO 和工件载入规则 LEFT 共同作用时最小, 也即说明了系统的等刀时间受工件流和刀具流共同影响。由图 6.5 可知: 在几类工件载入规则中, LEFT 规则与大部分刀具需求选择规则组合都可以得到较优的刀具交换次数, 这主要是因为 LEFT 规则充分考了工件流对刀具流的影响, 在机床选工序时, 优先选择向其它机床申请刀具数量最少的工序作为本机床的下一道待加工工序, 这就使得任意时刻系统中的刀具申请数量得到尽可能的控制, 从而能够显著地减少刀具交换次数, 减小刀具流负荷, 进而改善系统性能。

6.2.2 有故障的启发式调度运行结果

假设系统运行过程中发生机床故障和刀具破损, 其中设机床故障概率为 0.001, 刀具破损概率为 0.05, 机床的修理时间服从 7-30 的分钟的均匀分布。依据不同的工件载入规则和刀具需求选择规则得到的系统的各性能指标如图 6.7-6.11 所示。

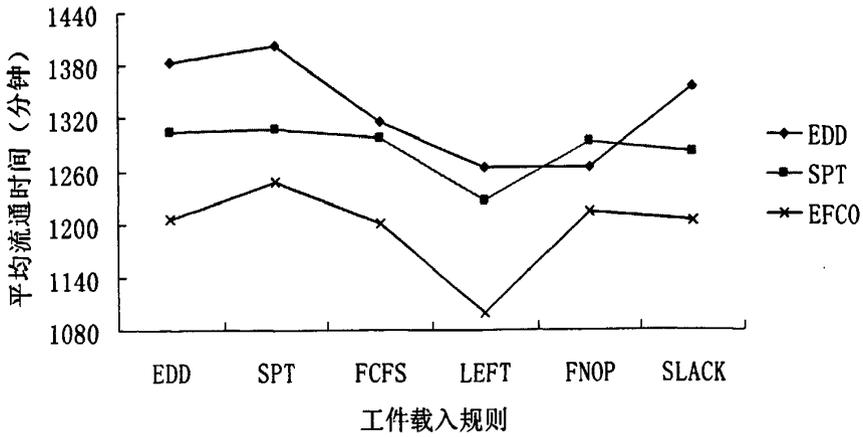


图 6.7 工序备刀下有故障平均流通时间

Fig.6.7 MFT with failure in case of procedure provisioning tools

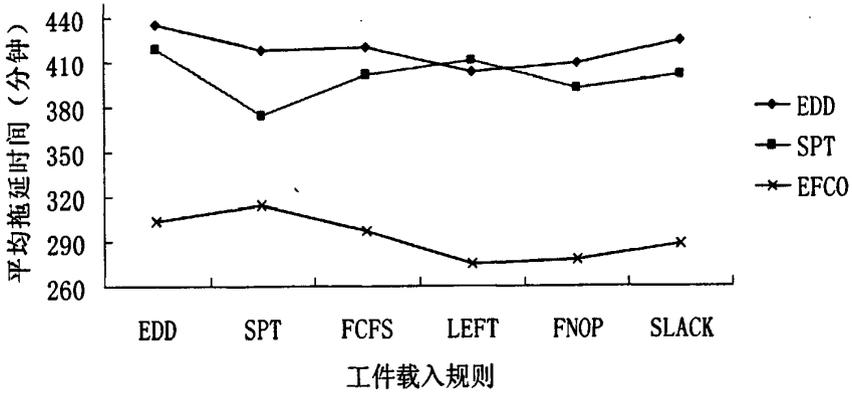


图 6.8 工序备刀下有故障的平均拖延时间

Fig.6.8 MTT with failure in case of procedure provisioning tools

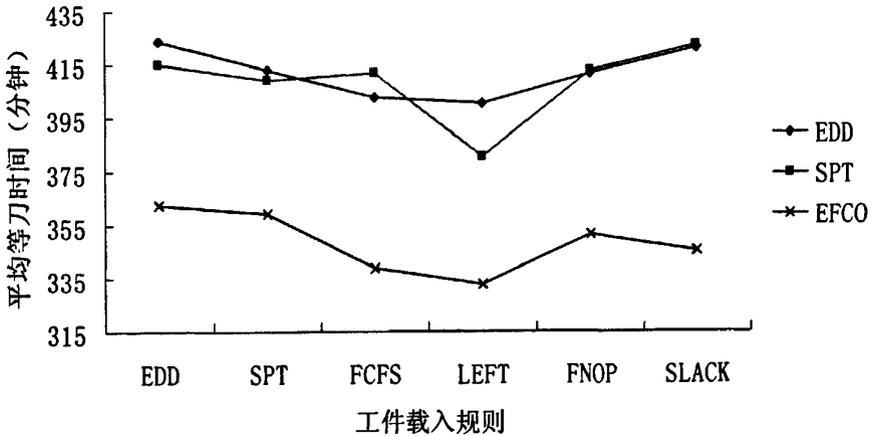


图 6.9 工序备刀下有故障的平均等刀时间

Fig.6.9 MWTT with failure in case of procedure provisioning tools

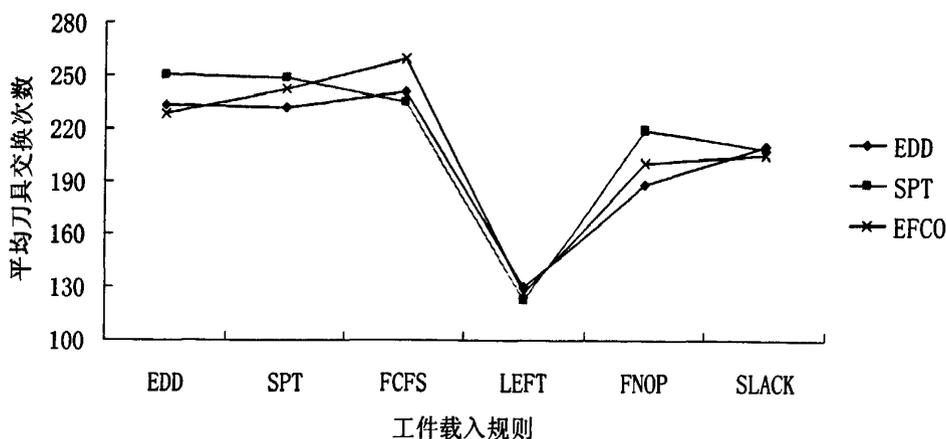


图 6.10 工序备刀下有故障的平均刀具交换次数

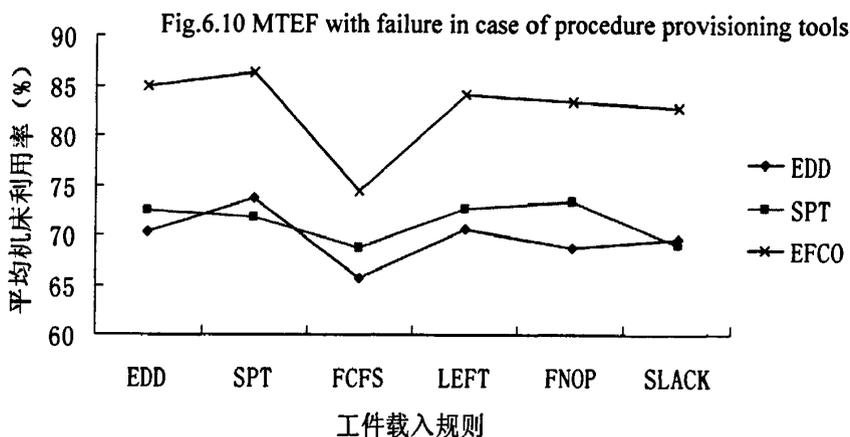


图 6.11 工序备刀下有故障的平均机床利用率

Fig.6.11 AMU with failure in case of procedure provisioning tools

为了分析有故障情况下不同的工件载入规则 PLR 和刀具需求选择规则 TRSR 对系统各性能指标的影响,采用双因素方差分析法进行方差分析,结果如表 6.3 所示。

表 6.3 工序备刀下有故障 PLR-TRSR 方差分析结果

Tab.6.3 Summary of ANOVA results with failure in case of procedure provisioning tools

系统性能指标	工序备刀下的 p 值		
	PLR	TRSR	Interac
MFT	0.0435	0.0031	0.8826
MTT	0.0413	0.0030	0.7921
MWTT	0.0390	0.0039	0.0416
MTEF	0.0035	0.0402	0.5941
AMU	0.0436	0.0042	0.8137

从表 6.3 可知,在考虑机床故障和刀具破损等故障情况时,工件载入规则和刀具需

求选择规则对系统各性能指标的影响与无故障时基本是一样的，也即对系统的平均流通时间，平均拖延时间和平均机床利用率等指标，刀具需求选择规则的影响更大一些，对平均等刀时间则受工件载入规则和刀具需求选择规则共同的影响，而对平均刀具交换次数指标，则受工件载入规则的影响更大一些。

从图 6.7-6.11 可知：在考虑故障时，系统的平均流通时间、平均拖延时间、平均等刀时间和平均刀具交换次数等指标都有了明显的增加，说明机床故障和刀具破损对系统的性能有明显的影 响，所以在车间调度中，必须对机床故障和刀具破损等情况进行故障检测，并且在调度算法中予以考虑。同时图 6.7-6.11 也说明了虽然调度过程中发生故障，但由于本文所提算法对故障的调度处理，工件加工过程仍协调进行，从而证明了工序备刀下刀具流无死锁调度算法的有效性。

表 6.4 中列出了无故障和有故障两种情况下获得较优的系统性能指标的工件载入规则和刀具需求选择规则组合。

表 6.4 工序备刀下较优的 PLR-TRSR 组合

Tab.6.4 Desirable operational policies in case of procedure provisioning tools

系统性能指标	工序备刀下的较优的 PLR-TRSR	
	无故障	有故障
MFT	EFCO-LEFT	EFCO-LEFT
MTT	EFCO-LEFT	EFCO-LEFT
MWTT	EFCO-LEFT	EFCO-LEFT
MTEF	*-LEFT; *-FNOP	*-LEFT; *-FNOP
AMU	EFCO-LEFT	EFCO-LEFT

注：*表示任意的刀具需求选择规则，PLR-TRSR 表示工件载入规则-刀具需求选择规则组合。

由表 6.4 可知，故障对调度规则的调度性能影响很小，也即 在无故障时能够获得较优系统性能指标的调度规则在有故障情况下同样可以获得较好的调度性能。

6.3 工步备刀下启发式调度运行结果及分析

本节主要针对系统运行中无故障和有故障两种情况，对 6 个工件载入规则和 3 个刀具需求选择规则生成的 18 个试验情形进行仿真运行并进行统计分析，以验证工步备刀下启发式调度算法的有效性 及评价各启发式调度规则的调度性能。其中，18 个试验情形，每一种情形重复试验 10 次，系统的各性能指标取 10 次试验的平均值。首先简要介绍一下各启发式调度规则。

1) 工件载入规则

工件载入规则 EDD、SPT、FCFS 和 SLACK 同工步备刀下的工件载入规则 EDD、SPT、FCFS 和 SLACK，而 Koulamas 规则为一种启发式规则，具体步骤如下：

i: 按照 SPT 排列工件;

ii: 从工件对列中选择要调度的工件 i , 然后将工件 i 的完工时间 d_i 与其紧邻的工件 j 的完工时间 d_j 比较, 若满足 $d_j \leq d_i$, 则调度工件 j 。

2) 刀具需求选择规则

工步加工时间最短 SOT (Shortest Operation Time): 该规则规定刀具首先分派给其虚拟机床队列中工步加工时间最短的机床。

先申请先服务 FCFS (First Come First Serve): 该规则规定先进入虚拟机床队列的机床先接受刀具服务。

剩余工步数最少 LOR (Least Operation Remaining): 该规则规定刀具首先分派给其虚拟机床队列中, 机床上在加工工序其剩余工步数最少的机床。

6.3.1 无故障的启发式调度运行结果

假设系统在运行过程中无机床故障和刀具破损发生, 依据不同的工件载入规则和刀具需求选择规则得到的系统的各性能指标如图 6.12-6.16 所示。

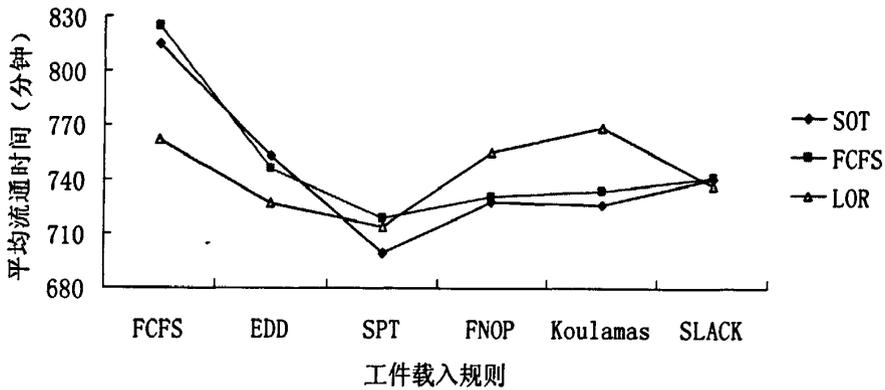


图 6.12 工步备刀下无故障的平均流通时间

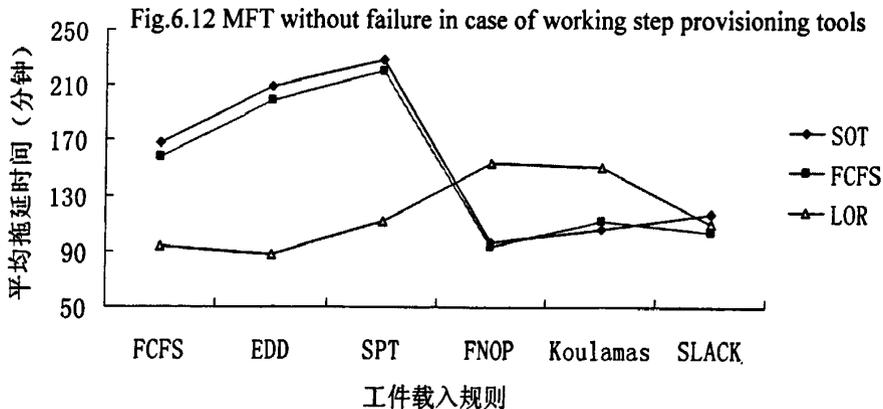


图 6.13 工步备刀下无故障的平均拖延时间

Fig.6.13 MTT without failure in case of working step provisioning tools

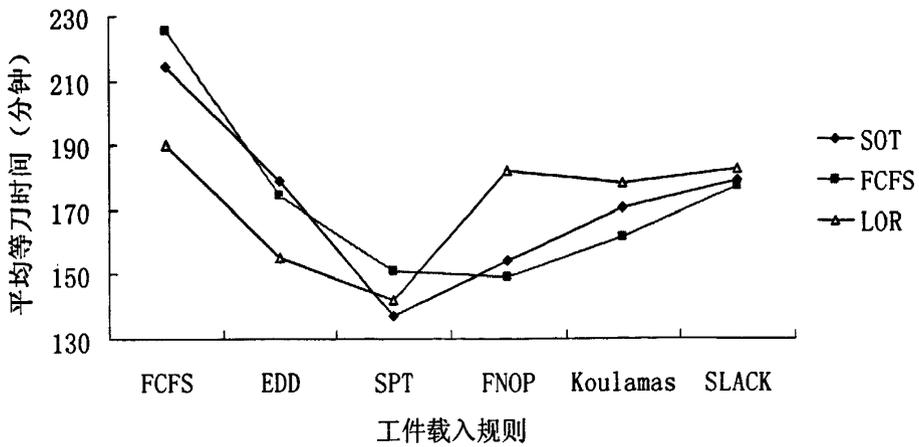


图 6.14 工步备刀下无故障的平均等刀时间

Fig.6.14 MWTT without failure in case of working step provisioning tools

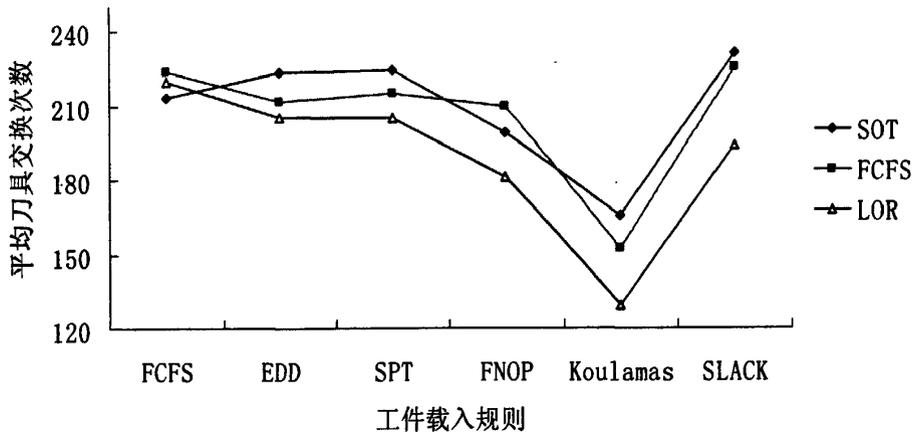


图 6.15 工步备刀下无故障的平均刀具交换次数

Fig.6.15 MTEF without failure in case of working step provisioning tools

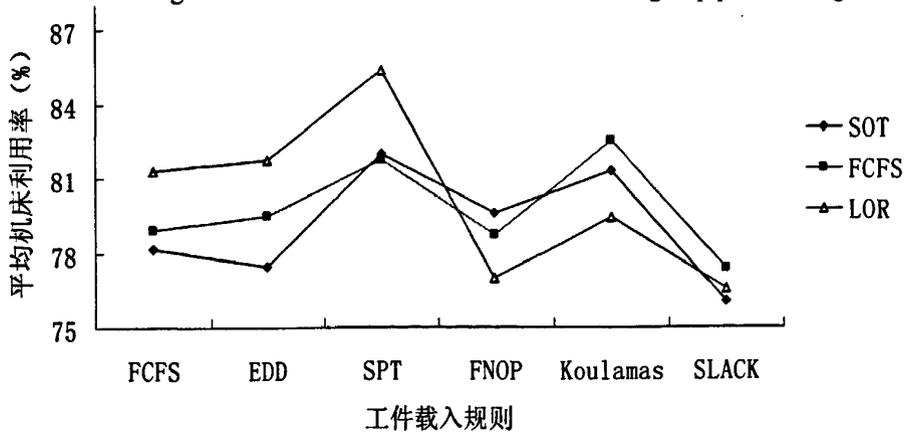


图 6.16 工步备刀下无故障的平均机床利用率

Fig.6.16 AMU without failure in case of working step provisioning tools

为了分析无故障情况下不同的工件载入规则和刀具需求选择规则对工步备刀情况的系统各性能指标的影响,采用双因素方差分析法进行方差分析,结果如表 6.5 所示。

表 6.5 工步备刀下无故障 PLR-TRSR 方差分析结果

Tab.6.5 Summary of ANOVA results without failure in case of working step provisioning tools

系统性能指标	工序备刀下的 p 值		
	PLR	TRSR	Interaction
MFT	0.0038	0.0412	0.8900
MTT	0.0015	0.0399	0.8206
MWTT	0.0081	0.0386	0.0360
MTEF	0.0036	0.0415	0.7010
AMU	0.0048	0.0431	0.7139

从表 6.5 可知,工件载入规则和刀具需求选择规则对系统的各性能指标都有影响,但相对于工件载入规则,刀具需求选择规则的影响小一些,这主要是由于在工步备刀前提下,工件在工序所需要的第一把刀具准备好后即可开始加工,加工完毕后,如果后续操作不再使用此把刀具即可释放,这致使刀具在大部分时间都是可得的,也即工步备刀思想大大提高了刀具资源的利用率,降低了刀具资源对加工过程的影响,故刀具需求选择规则对系统性能的影响小些。

由图 6.12, 6.14-6.16 可知, LOR 作为刀具需求选择规则与工件载入规则 FCFS 和 EDD 组合可以获得较好的系统平均流通时间,平均等刀时间、平均刀具交换次数和平均机床利用率,而 SOT 作为刀具需求选择规则与工件载入规则 SPT, FNOP, Koulamas 以及 SALCK 组合可以获得较好的调度性能,因此, SPT 作为工件载入规则和 SOT 作为刀具需求选择规则可以获得最优的平均流通时间,平均等刀时间和平均机床利用率。

由图 6.13 可知, LOR 作为刀具需求选择规则与工件载入规则 EDD, FCFS, SPT 组合可以获得较优的系统平均拖延时间,而 FCFS、SOT 作为刀具需求选择规则与工件载入规则 FNOP, Koulamas 和 SLACK 可以获得较好的调度性能,因此工件载入规则 EDD 和刀具需求选择规则 LOR 组合可以获得最优的系统平均拖延时间。

6.3.2 有故障的启发式调度运行结果

假设系统运行过程中发生机床故障和刀具破损,其中设机床故障概率为 0.001,刀具破损概率为 0.05,机床的修理时间服从 7-30 的分钟均匀分布。依据不同的工件载入规则和刀具需求选择规则得到的系统的各性能指标如图 6.17-6.21 所示。

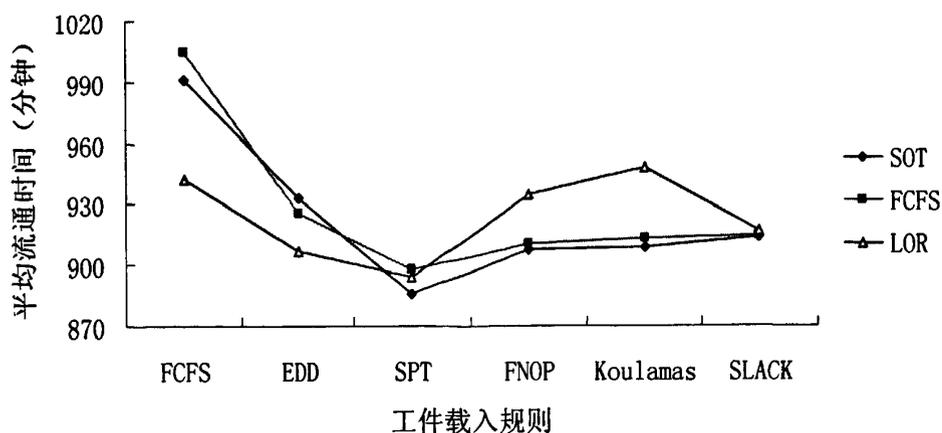


图 6.17 工步备刀下有故障的平均流通时间

Fig.6.17 MFT with failure in case of working step provisioning tools

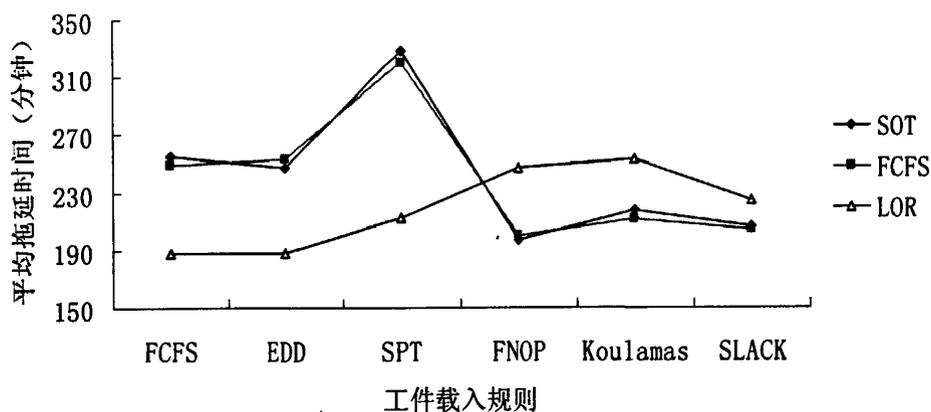


图 6.18 工步备刀下有故障的平均拖延时间

Fig.6.18 MTT with failure in case of working step provisioning tools

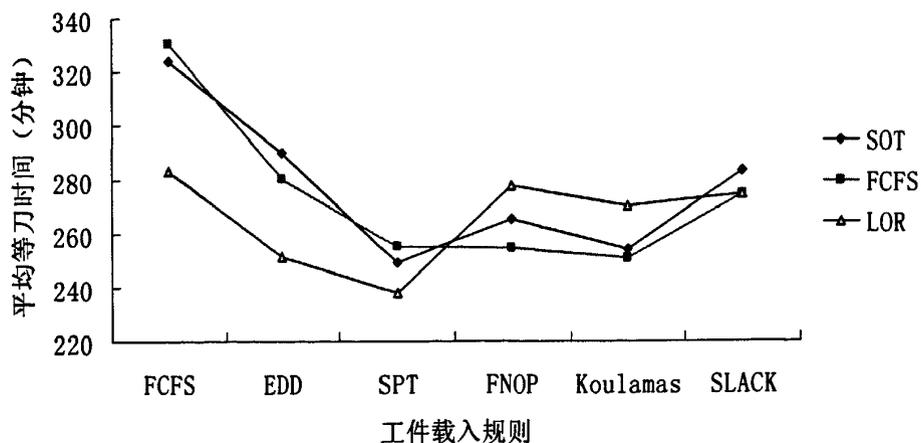


图 6.19 工步备刀下有故障的平均等刀时间

Fig.6.19 MWTT with failure in case of working step provisioning tools

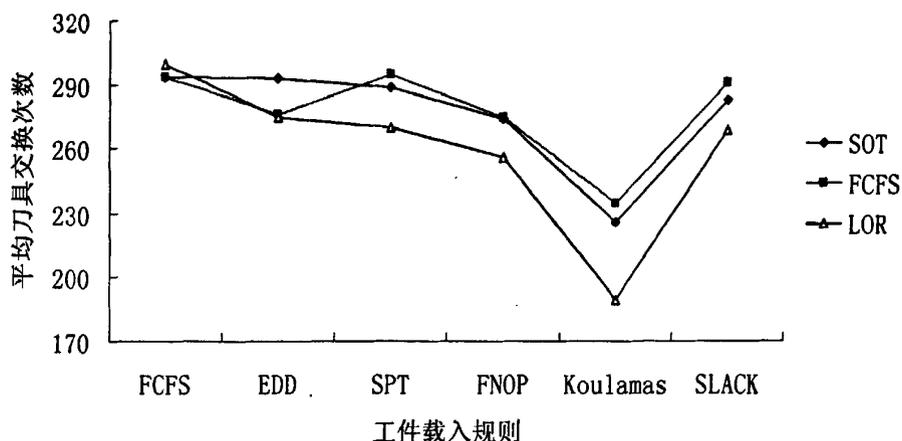


图 6.20 工步备刀下有故障的平均刀具交换次数

Fig.6.20 MTEF with failure in case of working step provisioning tools

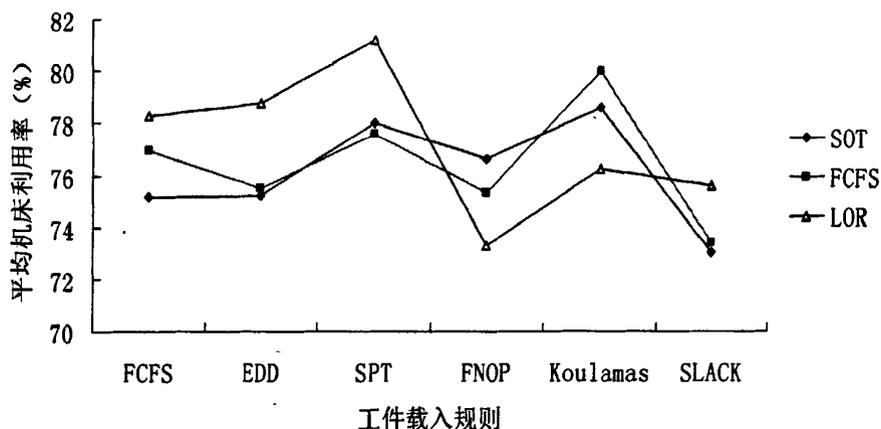


图 6.21 工步备刀下有故障的平均机床利用率

Fig.6.21 AMU with failure in case of working step provisioning tools

为了分析有故障情况下不同的工件载入规则和刀具需求选择规则对工步备刀下系统各性能指标的影响,采用双因素方差分析法进行方差分析,结果如表 6.6 所示。

表 6.6 工步备刀下有故障 PLR-TRSR 方差分析结果

Tab.6.6 Summary of ANOVA results with failure in case of working step provisioning tools

系统性能指标	工序备刀下的 p 值		
	PLR	TRSR	Interaction
MFT	0.0042	0.0491	0.8710
MTT	0.0021	0.0387	0.9911
MWTT	0.0079	0.0391	0.0323
MTEF	0.0037	0.0416	0.8104
AMU	0.0049	0.0411	0.9091

从表 6.6 可知, 在考虑机床故障和刀具破损等故障情况时, 工件载入规则和刀具需求选择规则对系统各性能指标的影响与无故障时基本一样, 也即对系统的各性能指标, 工件载入规则和刀具需求选择规则对其都有影响, 但相比于刀具需求选择规则, 工件载入规则的影响更大一些。

从图 6.17-6.21 不难发现, 与工序备刀情况相似, 当有故障情况发生时, 系统的平均流通时间、平均拖延时间、平均等刀时间和平均刀具交换次数等指标也都有了明显的增加, 但经过工步备刀下的无死锁调度算法对故障的处理, 工件仍能协调有序地得到加工。

表 6.7 中列出了无故障和有故障两种情况下获得较优的系统性能指标的工件载入规则和刀具需求选择规则组合。

表 6.7 工步备刀下较优的 PLR-TRSR 组合

Tab.6.7 Desirable operational policies in case of working step provisioning tools

系统性能指标	工步备刀下的较优的 PLR-TRSR	
	无故障	有故障
MFT	SPT-SOT	SPT-SOT
MTT	EDD-LOR	EDD-LOR
MWTT	SPT-SOT	SPT-SOT
MTEF	Koulamas-*	Koulamas-*
AMU	SPT-SOT	SPT-SOT

注: *表示任意的刀具需求选择规则, PLR-TRSR 表示工件载入规则-刀具需求选择规则。

由表 6.7 可知, 与工序备刀情况类似, 故障对调度规则的调度性能影响很小, 也在无故障时能够获得较优系统性能指标的调度规则在有故障情况下同样可以获得较好的调度性能。

6.4 工步备刀下优化调度运行结果及分析

本仿真实验主要用于测试所提出的工步备刀下的优化调度算法的有效性。测试案例中的待加工工件有5种类型, 共8个, 各工件编号分别为1-8, 加工工艺中所需刀具类型共6种, 其中刀具数量初始配置为: 1-5类刀具每类2把, 第6类刀具1把; 加工工艺参数如表6.8所示。优化过程中的双重遗传算法的基本参数如表6.9所示。

表6.8 工艺数据

Tab.6.8 Procedure parameters

工件类型	工序序号	加工机床号	工步序号(刀具类型号/刀具使用时间 min)	
			1	2
1	1	1	1/18	2/13
	2	3	3/19	2/20
	3	5	4/12	5/11
2	1	2	6/21	2/17
	2	4	3/15	5/25
	3	3	4/29	1/18
3	1	4	1/12	4/11
	2	1	5/30	3/12
4	1	5	3/20	5/24
	2	2	2/19	6/17
5	1	4	1/20	4/22
	2	1	5/18	3/18

表6.9 遗传算法的基本参数

Tab.6.9 Basic information of gegenetic algorithm

外层参数	N1=10	Pc1=0.8	Pm1=0.01	Counter=30
内层参数	N2=10	Pc2=0.8	Pm2=0.01	Counter=30

基于双重算法产生预调度方案，其工件加工顺序如图 6.22 所示，刀具分派顺序如表 6.10 所示。

表6.10 刀具破损前的刀具分派方案

Tab.6.10 Tool assignment scheme before tools breakage

刀具编号	刀具分派顺序	刀具编号	刀具分派顺序
t_1	$O_{211} \rightarrow O_{111} \rightarrow O_{332}$	t_7	$O_{511} \rightarrow O_{611} \rightarrow O_{432}$
t_2	$O_{312} \rightarrow O_{112} \rightarrow O_{222} \rightarrow O_{821} \rightarrow O_{721}$	t_8	$O_{212} \rightarrow O_{412} \rightarrow O_{122}$
t_3	$O_{711} \rightarrow O_{811} \rightarrow O_{221} \rightarrow O_{622}$	t_9	$O_{321} \rightarrow O_{421} \rightarrow O_{522} \rightarrow O_{121}$
t_4	$O_{231} \rightarrow O_{431}$	t_{10}	$O_{512} \rightarrow O_{331} \rightarrow O_{612} \rightarrow O_{131}$
t_5	$O_{712} \rightarrow O_{812} \rightarrow O_{422} \rightarrow O_{232} \rightarrow O_{132}$	t_{11}	$O_{322} \rightarrow O_{521} \rightarrow O_{621}$
t_6	$O_{311} \rightarrow O_{411} \rightarrow O_{822} \rightarrow O_{722}$		

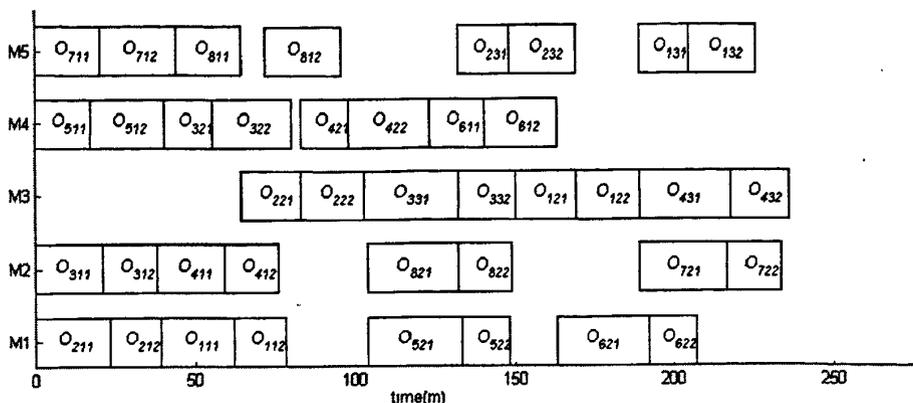


图 6.22 刀具破损前工件的加工顺序

Fig.6.22 Machining sequence of parts before tools breakage

在 $t=70$ 分钟时, 机床 M_1 加工 O_{112} 操作时候, 所用的 2 号刀具发生破损, 故双重遗传算法对剩余未加工的工序进行再调度, 产生再调度方案中工件的加工顺序如图 6.23 所示, 再调度后刀具的分派顺序如表 6.11 所示。

表 6.11 刀具破损后的刀具分派方案

Tab.6.11 Tool assignment scheme after tools breakage

刀具编号	刀具分派顺序	刀具编号	刀具分派顺序
t_1	$O_{611} \rightarrow O_{332}$	t_7	O_{432}
t_2	X (破损)	t_8	$O_{222} \rightarrow O_{112} \rightarrow O_{721} \rightarrow O_{122} \rightarrow O_{821}$
t_3	$O_{421} \rightarrow O_{121} \rightarrow O_{522}$	t_9	O_{622}
t_4	$O_{331} \rightarrow O_{431}$	t_{10}	$O_{612} \rightarrow O_{231} \rightarrow O_{131}$
t_5	$O_{521} \rightarrow O_{232} \rightarrow O_{132}$	t_{11}	$O_{422} \rightarrow O_{621}$
t_6	$O_{722} \rightarrow O_{822}$		

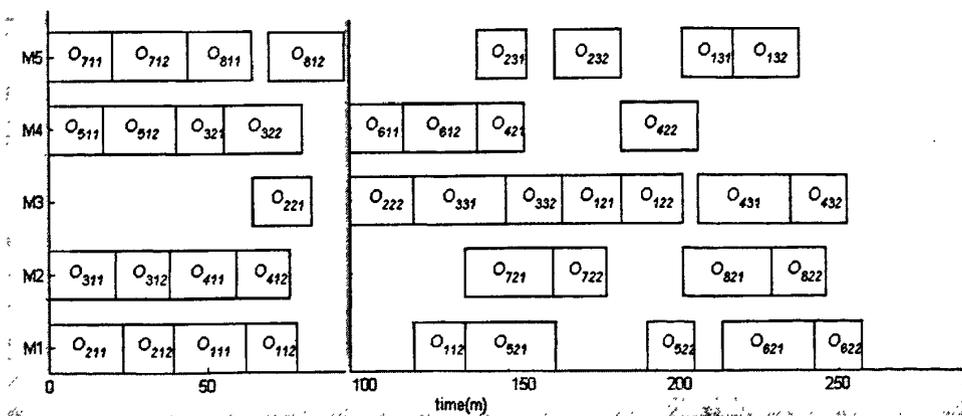


图 6.23 刀具破损后工件的加工顺序

Fig.6.23 Machining sequence of parts after tools breakage

综上所述可知,即使在加工过程中,刀具发生了破损,但是由于本调度算法对刀具失效的调度处理,失效刀具的任务被同类型的其他刀具迅速承担,工件的加工过程仍较为协调,从而证明了本文所提出的基于双重遗传算法的准静态优化再调度方法是健壮而有效的。

类似的,依据本文仿真实例的生成方法,随机生成 10 个仿真实例,得到双重遗传算法下的各性能指标的平均值如表 6.12 所示。

表 6.12 基于双重遗传算法的系统性能指标

Tab.6.12 Performance index based on double layer genetic algorithm

刀具破损率	MFT	MTT	MWTT	MTEF	AMU
0	650.3	68.0	115.4	152.3	87.8
0.025	741.6	132.7	165.7	171.1	84.6
0.05	886.2	192.4	235.4	215.9	82.7

6.5 各算法的综合评价

主要针对各算法在无故障和有故障两种情况下,系统的各性能指标进行对比分析,做出综合评价。

无机床故障和刀具破损发生时,各算法下系统的性能指标如表 6.13 所示。其中:A1 表示工序备刀下刀具流无死锁的启发式调度算法;A2 表示工步备刀下刀具流无死锁的启发式调度算法;D-GA 表示工步备刀下基于双重遗传算法的准静态再调度算法。

表 6.13 无故障时不同算法的性能指标

Tab. 6.13 Performance index of different algorithms without failure

系统性能指标	调度算法		
	A1	A2	D-GA
MFT	887.7	700.0	650.0
MTT	161.3	80.6	68.9
MWTT	189.5	137.3	115.4
MTEF	83.1	129.7	152.3
AMU	81.5	86.5	87.8

为了比较各算法的调度性能,基于表 6.13 的数据绘制不同算法的对比柱状图如图 6.24 所示。

由图 6.24 可见,与工序备刀下的启发式调度(A1 算法)相比,当不考虑机床故障和刀具破损等情况时,工步备刀下(A2 算法和 D-GA 算法)系统的平均流通时间,平均拖延时间及平均等刀时间等性能指标明显改善,但刀具交换次数却有明显的增加,这一方面验证了本文所提的工步备刀思想的正确性和可行性,另一方面也表明工

步备刀思想是以刀具交换次数的增加为代价来换取系统其它性能指标的提高。

同样在工步备刀前提下，针对无故障情况，分析结果表明，不同的调度方法对应的系统性能也有较大不同，工步备刀下的优化调度性能显著优于启发式调度。例如启发式调度下的平均流通时间和等刀时间分别为 700min、137.3min，而优化调度下相应的指标为 650min、115.4min，降低幅度分别达 7.14%和 16.01%。不过，平均刀具交换次数也发生了较大的变化，例如启发式调度下为 129.7 次，而优化调度下有了较为明显的增加，为 152.3 次，增幅达 17.4%。与前文所述类似，显然优化调度也是以刀具交换次数的增加为代价来换取系统其它性能指标的改善。

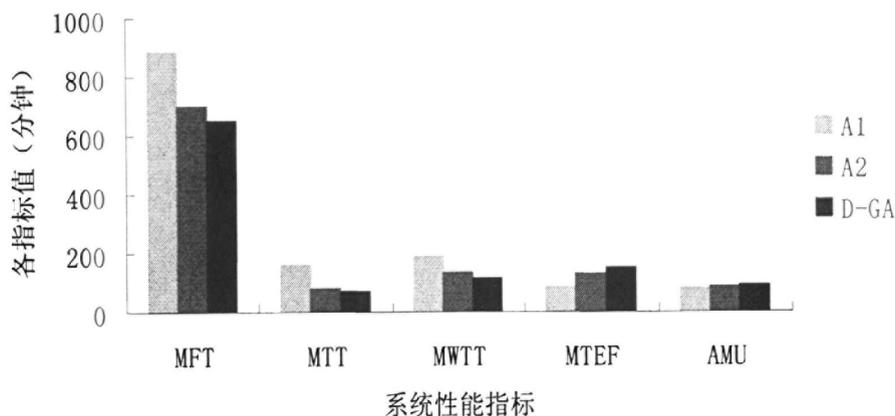


图 6.24 无故障时不同算法的性能指标

Fig.6.24 Performance index of different algorithms without failure

当系统加工过程中考虑机床故障和刀具破损时，系统的性能指标如表 6.14 所示。

表 6.14 有故障时不同算法的性能指标

Tab. 6.14 Performance index of different algorithms with failure

系统性能指标	算法		
	A1	A2	D-GA
MFT	1098.5	895.3	886.1
MTT	250.8	180.0	192.6
MWTT	332.5	237.9	235.4
MTEF	119.2	188.1	215.9
AMU	79.5	83.5	82.7

为了比较工序备刀和工步备刀的调度性能，基于表 6.14 的数据绘制不同算法的对比柱状图如图 6.25 所示。

由图 6.25 可见，在调度过程中，若引入机床故障和刀具破损等异常事件，工步备刀下无论是启发式调度结果还是优化调度结果，仍然都优于工序备刀的调度性能，这

说明工步备刀思想在车间动态调度情况下是更为可行而有效的。

在工步备刀前提下，在考虑实际生产中可能发生的机床故障和刀具破损后，启发式调度结果和优化调度结果相差不大，这主要是因为此类工况下基于双重遗传算法的优化再调度需要多次进行重调度，因而只能保证在分段调度中寻找近优解，而不能实现全局意义上的近优解。

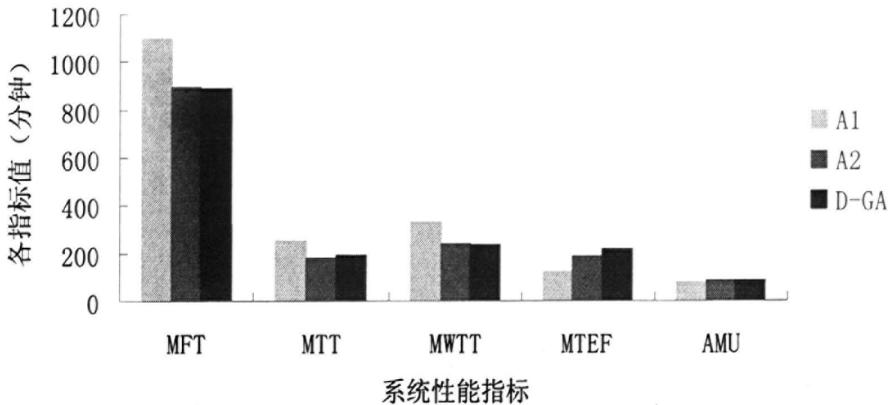


图 6.25 有故障时不同算法的性能指标

Fig.6.25 Performance index of different algorithms with failure

综上所述可见，在车间调度过程中，无论是否发生机床故障和刀具破损等情况，除了换刀次数指标，工步备刀前提下的调度性能都明显优于工序备刀情况。而对工步备刀前提，若不考虑机床故障和刀具破损等故障情况，以系统完工时间为目标的基于双重遗传算法的准静态优化再调度方法在工件的流通时间及等刀时间上明显优于启发式调度，但是在换刀次数上有明显的增加；而若考虑机床故障和刀具破损等故障情况，则基于双重遗传算法的优化调度结果与启发式调度结果相差不大。

6.6 本章小结

本章主要对各调度算法进行仿真试验并对其运行结果进行分析。首先基于工序备刀和工步备刀，针对无故障和有故障两种情况，分别对6个工件载入规则和3个刀具需求选择规则生成的18个仿真试验进行仿真运行并进行统计分析，评价了各启发式调度规则的调度性能；其次基于工步备刀，对基于双重遗传算法的准静态再调度算法进行仿真试验，试验中考虑了刀具突然破损等情况，以验证算法对突发事件的响应。最后对三种算法下系统各性能指标进行了比较和综合评价，从分析结果可知，本文所建立的工序备刀下的启发式调度算法和工步备刀下的启发式调度算法及优化调度算法是可行且有效的，为FMS的设计和运行提供了依据，为实际生产中FMS性能的充分发挥奠定了基础。

结 论

本文对 FMS 中刀具流动态调度进行了理论分析和仿真研究。在深入分析国内外刀具流动态调度研究现状的基础上,针对 FMS 生产中工序备刀与工步备刀两种情况下的刀具流调度问题,分别进行了基于“仿真+规则”的启发式调度研究和基于智能搜索算法的优化调度研究,建立了适用的刀具流调度策略和算法,为充分发挥系统生产能力,提高系统生产效率提供了理论技术基础。同时,对刀具调度中的关键技术问题--刀具流死锁进行了深入的分析,准确地揭示了刀具流死锁的本质属性,丰富和完善了刀具流调度理论。

论文取得的主要创新性成果如下:

(1) 深入研究了柔性制造系统中的刀具流建模问题。利用面向对象技术和 Petri 网理论,针对工序备刀与工步备刀两种情况,分别建立了工件流和刀具流综合调度仿真模型,在该模型中不仅考虑了刀具寿命的影响,同时还把刀具破损等故障情况包括进来,为后续的刀具流启发式调度研究奠定了基础。

(2) 对工序备刀下的刀具流死锁问题进行了进一步的研究和完善。针对考虑刀具寿命的刀具流,采用图论方法,建立了扩展的刀具申请分配图。提出了刀具流死锁检测算法和死锁避免策略,构建了刀具分派的启发式调度规则,给出了刀具流无死锁的启发式调度算法,解决了工序备刀下的刀具流死锁避免和刀具实时分派问题。

(3) 针对工步备刀下的刀具流死锁问题进行了研究。为工步备刀下的刀具流建立了刀具流状态演变图,给出了刀具流死锁状态和无死锁不安全状态的定义,提出了一种刀具流死锁避免策略,并设计了相应的刀具流无死锁启发式调度算法,解决了工步备刀下的刀具流死锁避免和刀具实时分派问题。

(4) 研究了刀具流优化调度问题。采用整数规划方法,针对工步备刀下的刀具流调度建立了数学模型,提出了一种双重遗传算法对模型进行优化求解。针对刀具破损等突发事件下的再调度问题,进一步提出了基于双重遗传算法的准静态再调度方法,有效地解决了工步备刀下的刀具实时分派问题。

今后研究工作中还需要关注的问题是:

(1) 进一步提高所提出的双重遗传优化算法的运行性能,如寻求全新的染色体编码方法和更合适的遗传操作算子以及采用并行遗传等,提高算法的搜索能力。

(2) 本文所取得的再调度结果仍属局部优化解,应对再调度问题作进一步深入研究,力争提高优化解的质量。

(3) 本文解决刀具流死锁的方案主要从控制的角度着手,如果能够从调度的角度出发来处理死锁,可以为系统的控制节省很多工作。

参考文献

- [1] 钱晓龙,唐立新,刘文新.动态调度的研究方法综述.控制与决策,2001,16(2):141-145 页
- [2] Jackson J R. Simulation research on job shop production. Naval Res Log Quart, 1957,4(3):287-295P
- [3] 胡立德,张秉铝,王时龙.FMS 动态调度的应用.机床,1993,(12):25-28 页
- [4] 李宏,王宁生.FMS 调度软件自动生成系统原理及其结构.组合机床与自动化加工技术,1995,(7):27-33 页
- [5] K.Ravi raju,O.V.Krishnaiah Chetty.Priority Nets for Scheduling Flexible Manufacturing Systems. Journal of Manufacturing Systems,1996,12(4):326-339P
- [6] E.Arjona-Suarez,E.Lopez-Mellado.Synthesis of Coloured Petri Nets for FMS Task Specification.International Journal of Robotics and Automation,1996,11(3): 111-119P
- [7] DooYong Lee, Frank DiCesare. Scheduling Flexible Manufacturing Systems Using Petri Nets and Heuristic Search. IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, 1994,10(2):123-132P
- [8] 吴智铭.散件调度和监控中的 FMS 仿真器:第四届现代制造技术国际学术讨论会论文集,116-121 页.1993[C].上海,中国
- [9] 胡泓,陈剑,丁国骏.递阶着色 Petri 网在面向对象设计中的应用.上海交通大学学报,1999,(11):1422-1424 页
- [10] Federico Della Croce , Roberto Tadei, Gruseppe Volta. A Genetic Algorithm for The Job Shop Problem . Computers Ops Res,1995,22(1):15-24P
- [11] 王志亮,汪惠芬,张友良.动态 Job-Shop 调度问题的一种自适应遗传算法.中国机械工程,2004,(11):995-999 页
- [12] Magdy M.Abdelhameed , Farid A.Tolbah. Design and Implementation of a Flexible Manufacturing Control System Using Neural Network. International Journal of Flexible Manufacturing Systems,2002,(14):263-279P
- [13] 宋本基,喻宏斌,郑鹏,崔维娜.FMS 故障调度问题的研究及仿真.哈尔滨工程大学学报,2000,21(08):56-59 页
- [14] Sarin S C, Salgame R.A. knowledge-based system approach to dynamic scheduling. Knowledge-based system, 1989,(08):173-206P
- [15] 萧蕴诗,李曦等.基于规则的 FMS 实时调度专家系统.组合机床与自动化加工技

- 术,1995,(11):36-40 页
- [16] 刑善松,陈炳森.实用 FMS 动态调度专家系统设计.组合机床与自动化加工技术,1996,(3):19-22 页
- [17] 周炳海,蔡建国.基于定向搜索的 FMS 调度优化算法.组合机床与自动化加工技术,2001,(10):1-4 页
- [18] S.A.reveliotis, M.A.Lawley , P.M.Ferreira. Structural control policies for flexible manufacturing systems:Proc 11th Int Conf.Computer-aided Prod,255-260P.1995[C]
- [19] S.A.Reveliotis, P.Ferreira. Deadlock avoidance policies for automated manufacturing cells robot. Automat,1996,(12):845-857P
- [20] Sang Chan Park, Narayan Raman, Michael J.Shaw. Adaptive Scheduling in Dynamic Flexible Manufacturing Systems: A Dynamic Rule Selection Approach. IEEE Transactions on Robotics and Automation,1997,13(4):486-495P
- [21] Chang-Yong Song, Hark Hwang, Yeong-Dae Kim. Heuristic Algorithm for the Tool Movement Policy in Flexible Manufacturing Systems. Journal of Manufacturing Systems,1995,14(3):160-168P
- [22] 朱世和,高东.实现刀具优化配置的方法研究.组合机床与自动化加工技术,1995,(4):31-35 页
- [23] P.H.Koo, J.M.A.Tanchoco, J.J.Talavage. Tool requirements in manufacturing systems under dynamic tool sharing: In Proc.20th Int. Conf.Comput. Industr.Eng, 1271-1274P .1996[C]
- [24] Ashok Kumar, Zubair Mohamed, Jaideep Motwani, Mohamed Youssef. A Dynamic Tool Requirement Planning Model for Flexible Manufacturing Systems. International Journal of Flexible Manufacturing Systems. 1997,9(4):307-342P
- [25] P.H.Koo, J.M.A.Tanchoco, J.J.Talavage. Estimation of tool requirements in single-stage multimachine systems. Int. J. Prod. Res,1998,36(4):1699 -1713P
- [26] Hong-Bae Jun, Yeong-Dae Kim. Heuristics for a Tool Provisioning Problem in a Flexible Manufacturing System with an Automatic Tool Transporter. Robotics and Automation,1999,15(3):488-496P
- [27] Yeong-Dae Kim, Geun-Cheol Lee, Seong-Kook Choi, Seung-Kil Lim. Tool requirements planning in a flexible manufacturing system with an automatic tool transporter. Robotics and Automation, 2001,17(6):795-804P
- [28] Vrečer, G., Cus, F.. Planning of tool supply to a group of machines. Journals of Material Processing Technology,1998,133(2):214-217P
- [29] 王解法,冯祖仁等.柔性制造系统刀具需求规划问题的研究.组合机床与自动化加

- 工技术,2003,(9):21-24 页
- [30] NOELA M, SODHI M S, LAMOND B F. Tool planning for a lights-out machining system. *Journal of Manufacturing Systems*,2007, 26(3-4):161-166P
- [31] 陈进,段黎明.FMS 的刀具流在线优化控制策略.制作技术与机床,1999,(06):40-43 页
- [32] Leung LC, Hui YV. Dynamic management of cutting tools for flexible and quality machining. *Int J Prod Res*, 2000,38(14):385-401P
- [33] Akturk MS, Ozkan S. Integrated scheduling and tool management in flexible manufacturing systems. *Int J Prod Res*, 2001,39(12):697-722P
- [34] Chen J, Chen FF. FMS loading in a tool-shared environment with partially grouped machines: IEE Proceedings of the Industrial Engineering Research Conference, 2002[C].Orlando, FL,USA
- [35] 李宏.多Agent在刀具调度中的应用.工具技术,2005,(12): 58- 60页
- [36] 王用超,汪永超,余显芳,张冬妮.基于蚁群算法的多Agent合同网协议的刀具调度.组合机床与自动化加工技术,2010,(06):109-112页
- [37] P.Tomek.Tooling strategies related to FMS management. *FMS Mag*, 1986,(4):102-107P
- [38] Chen-Hua Chung.Planning Tool Requirements for Flexible Manufacturing Systems.*Journal of Manufacturing Systems*,1998,10(6):476-483P
- [39] 张培忠.柔性制造系统[M].北京:机械工业出版社,1998
- [40] C.Zhou, R.A.Wysk. Tool Management for Computer Integrated Manufacturing:Proc Of 1989 Intl. Ind. Eng. Conf. And Mfg. And Productivity Symposium,603-607P. 1989[C]
- [41] A.E.Gray, A.Seidmann, K.e.Stecke. Tool Management in Automated Manufacturing: A Tutorial. Proc. Of 3rd ORSA/TIMS Conf. On Flexible Mfg. Systems: Operations Research Models and Applications, 93-98P. 1989[C]
- [42] A.S.Carrie, D.T.Perera. Work Scheduling in FMS under Tool Availability Constraints. *International Journal of Production Research*,1986,(24): 1299-1308P
- [43] R.Suri, C.K.Whitney. Decision Support requirements in Flexible Manufacturing. *Journal of Manufacturing Systems*,1984,3(1): 61-69P
- [44] Nikos C.Tsourveloudis, Yannis A.Phillis. Manufacturing Flexibility Measurement: A Fuzzy Logic Framework. *Robotics and Automation*,1998,14(4):513-523P
- [45] 欧阳普仁.FMS 刀具管理系统体系结构研究.南京理工大学学报,1996, 20(3): 273-276 页

- [46] 邹来龙,刘雄伟,袁哲俊.FMS 刀具流系统的规划和调度仿真.组合机床与自动化加工技术,1988,(09):28-34 页
- [47] G.J.C.Gaalman, W.M.Nawijn. Tool sharing in parallel part production. Int. J. Production Economics,1996, (46-47):521-533P
- [48] ZHANG Xiaoming,FUJII Susumu, KAIHARA Toshiya.A Study on the Tool Sharing Policy in FMS with Tool Center. JSME international journal. Series C,Mechanical systems, machine elements and manufacturing,2003,46(1): 03-15 ,24-32P
- [49] 赵中敏, 欧阳小安. 柔性制造系统的刀具动态调度管理系统.中国设备工程,2009,(3): 13-15 页
- [50] MATTA A,TOLIO T,TONTINI F.Tool management in flexible manufacturing systems with network part program.International Journal of Production Research, 2004,42(17):3707-3730P
- [51] 杨岳.FMS 刀具管理理论与方法的研究.重庆大学博士学位论文.1991
- [52] 赵 葛 霄 .FMS 环境下的刀具管理系统 .组合机床与自动化加工技术,2000,(09):45-47 页
- [53] 欧阳普仁,王拴虎等.FMS 分配与调度策略研究.组合机床与自动化加工技术,1998,(07):24-27 页
- [54] 李 玲 , 范 牧 昌 .FMS 刀具管理系统的研究 .华东船舶工业学院学报, 2000,14(05):50-53 页
- [55] 白传悦,张斌.柔性制造系统中的刀具流的管理.物流技术,2003,(04):24-25 页
- [56] 苏加国,邹福生.FMS 中刀具优化管理的数学建模与分析.昆明理工大学学报,2002,(04):38-41 页
- [57] 林天义,严新明.集成环境下的刀具调度管理系统.组合机床与自动化加工技术.1995(4):26-30 页
- [58] Suresh K.Khator, Lawrence C.Leung. Intermediate Tool Requirement Planning for FMS. Journal of Manufacturing Systems,1995,3(01):9-19P
- [59] Kwasi Amoako-Gyampah, Jack R.Meredith. A Simulation Study of FMS Tool Allocation Procedures. Journal of Manufacturing Systems,1996,15(06):419-431P
- [60] C. Reddy, O.V.K. Chetty, D. Chaudhuri. A Petri net based approach for analyzing tool management issues in FMS. International Journal of Production Research. 1992, 30(06):1427-1446P
- [61] K.BARKAOUI, I.BEN ABDALLAH. Modeling and Performance Evaluation of Tool Sharing Management in FMS Using Stochastic Petri Nets:Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 282-288P. 1993[C]

- [62] 毕诸明,朱岩等.FMS 运控软件调试环境中的刀具流死锁的检测.组合机床与自动化加工技术,1996,(01):19-22 页
- [63] F.F.Chen, J.Chen. Performance Modeling and Evaluation of Dynamic Tool Allocation in Flexible Manufacturing Systems using Coloured Petri Nets: An Object-Oriented Approach. The International Journal of Advanced Manufacturing Technology,2003,21(02):98-109P
- [64] 舒海生.柔性制造系统工件流和刀具流动态调度研究.哈尔滨工程大学学位论文, 2005
- [65] Housni Djellab, Khaled Djellab, Michel Gourgand. A new heuristic based on a hypergraph representation for the tool switching problem. International Journal of Production Economics,2000,46(1-3): 165-176P
- [66] GEBRAEEL N Z, LAWLEY M. Deadlock detection, prevention, and avoidance for automated tool sharing systems. Robotics and Automation,2001,(17):342-356P
- [67] LEE.C.S, KIM.S.S, CHOI.J.S. Operation sequence and tool selection in flexible manufacturing systems under dynamic tool allocation. Computers & Industrial Engineering,2003, 45(01): 61-73P
- [68] T.S.Atan, R. Pandit. Auxiliary tool allocation in flexible manufacturing systems. Eur. J. Oper. Res,1996,(89):642-659P
- [69] Atmani.A, Lashkari.RSA. Model of machine tool selection and operation allocation in FMS. International Journal of Production Research,1998,(36):39-49P
- [70] Gamila.M.A, Motavalli.SA. Modeling technique for loading and scheduling problems in FMS. Robotics and Computer-Integrated Manufacturing,2003,(19): 45-54P
- [71] S.G. Ponnambalam, Low.Seng.Kiat. Solving Machine Loading Problem in Flexible Manufacturing Systems Using Particle Swarm Optimization:World Academy of Science, Engineering and Technology,2008[C]
- [72] Rahul.Swarnkar, M.K.Tiwari. Modeling machine loading problem of FMSs and its solution methodology using a hybrid tabu search and simulated annealing-based heuristic approach.Robotics and Computer-Integrated Manufacturing, 2004,(20): 199-209P
- [73] S.Bilgin, M.Azizoğlu. Capacity and tool allocation problem in flexible manufacturing systems. Journal of the Operational Research Society,2006,(06): 670-681P
- [74] Chang-Yong.Song, Hark Hwang, Yeong-Dae.Kim. Heuristic Algorithm for the Tool

- Movement Policy in Flexible Manufacturing Systems. *Journal of Manufacturing Systems*,1995,14(3):160-168P
- [75] Arun.S.Kashyap, Suresh K. Khator. Analysis of tool sharing in an FMS: A simulation study sloving tool allocation problem in flexible manufacturing system. *Computers & Industrial Engineering*,1996,30(01):137-145P
- [76] H.K.Ro, Y.D.Kim. Due-date based loading and scheduling methods for a flexible manufacturing system with an automatic tool transporter. *Int. J. Prod. Res.*, 1997,(35): 2989–3003P
- [77] Choong Soo Lee, Hyun Jong Kim. A Part Release considering Tool Scheduling and Dynamic Tool Allocation in Flexible Manufacturing Systems: 2006IEEE
- [78] Y. Fathi, K.W. Barnette. Heuristic procedures for the parallel machine problem with tool switches. *Int J Prod Res*, 2002,40(1):151–164P
- [79] Kwasi Amoako-Gyampah, Jack R. Meredith, T. Tolio, S. Toma. A simulation study of FMS tool allocation procedures. *Journal of Manufacturing Systems*,1996, 15(6):419-431P
- [80] Ozbayrak M., Turker, A.K.,Pisman, M.Part And Tool Flow Management In Multi-cell Flexible Manufacturing System: Simulation Conference, Proceedings of the 1997 Winter,1997[C]
- [81] BUYURGAN N, SAYGIN C, KILIC S E. Tool allocation in flexible manufacturing systems with tool alternatives. *Journal of Manufacturing Systems*,2004,20(4): 341-349P
- [82] SURESH K N, SRIDHARAN R. Simulation modeling and analysis of tool sharing and part scheduling decisions in single-stage multimachine flexible manufacturing systems. *Robotics and Computer-Integrated Manufacturing*, 2007,23(4): 361-370P
- [83] ZHANG.Xiaoming, FUJII Susumu, KAIHARA Toshiya. Evaluation of Tool Allocation Strategies in Flexible Manufacturing System. *JSME international journal. Series C, Mechanical systems, machine elements and manufacturing*, 2005,48(01):37-45P
- [84] SURESH K N, SRIDHARAN R. Performance of a flexible manufacturing system operating under part movement policy and tool movement policy:Simulation modelling and analysis: *Industrial Engineering and Engineering Management* 2008, IEEM 2008, 1754 – 1758P .2008[C]
- [85] SURESH K N, SRIDHARAN R. Simulation modeling and analysis of part and tool flow control decision in a flexible manufacturing system. *Robotics and*

- Computer-Integrated Manufacturing,2009, 25(4-5):829-838P
- [86] 席光辉.应用遗传算法研究 FMS 刀具需求模型.现代制造工程,2006,(06):69-70 页
- [87] MELTEM DENIZEL-SIVRI, S.SELCUK ERENGUC.A branch and bound procedure for setup problems in flexible manufactruing systems. Production and Operations Management,1993, 2(04) :289-305P
- [88] A. Grieco, Q.Semeraro, T.Tolio, S.Toma. Simulation of tool and part flow in fms. Int. J. Production Res,1995, 33(03):643-658P
- [89] Dario Pacciarelli. Loading part and tool in a flexible manufactruing system.1998[OL]. <http://www.inf.uniroma3.it/research/tech-rep/dia-3/>
- [90] M. SELIM AKTURK. An exact tool allocation a approach for CNC machines. INT. J. COMPUTER INTEGRATED MANUFACTURING,1999,12(02):129-140P
- [91] Selin Özpeynirci, Meral Azizoğlu. Bounding approaches for operation assignment and capacity allocation problem in flexible manufacturing systems. Computers & Operations Research,2009,36(09):2531-2540P
- [92] Selin.Özpeynirci,Meral.Azizoğlu.Capacity allocation problem in flexible manufactureing systems: branch and bound based approaches. International Journal of Production Research, 2009,47(21):5941-5958P
- [93] T. Tsukada, K. Shin. Distributed tool sharing in flexible manufacturing systems. Robot Automat,1998,(04):379-389P
- [94] R.Rai, S.Kameshwaran, M.K.Tiwari. Machine tool selection and operation allocation in FMS: solving a fuzzy goal-programming model using a genetic algorithm. Int J Prod Res,2002,40(03):641-665P
- [95] ZHANG Xiaoming, FUJII Susumu, KAIHARA Toshiya. A Study on the Tool Sharing Policy in FMS with Tool Center. JSME international journal. Series C, Mechanical systems, machine elements and manufacturing,2003,46(01):24-32, 03-15P
- [96] Felix T.S. Chan, Rahul Swarnkar. Ant colony optimization approach to a fuzzy goal programming model for a machine tool selection and operation allocation problem in an FMS. Robotics and Computer-Integrated Manufacturing,2006,(22): 353-362P
- [97] T.Prabakaran, P.R.Nakkeeran, N.Jawahar. Sequencing and scheduling of job and tool in a flexible manufacturing cell. The International Journal of Advanced Manufacturing Technology,2006,29(7-8):729-745P
- [98] 徐玲,杨丹,王时龙,聂建林.基于进化神经网络的刀具寿命预测.计算机集成制造系统,2008,1(14):28-34 页

- [99] M.Fanti,B.Maione,S.Mascolo,B.Turchiano.Event-based feedback control for deadlock avoidance in flexible production systems.IEEE Trans.Robot. Automat,1997,13(1):347-363P
- [100] 黄忠华.自动制造系统中的无死锁调度与控制.上海交通大学博士论文,2007
- [101] 韩赞东,李基范.FMS 中资源共享导致的死锁状态及其避免方法.中国机械工程,2004,15(10):887-891 页
- [102] Widodo Sulistyono,Mark A.Lawley.Deadlock Avoidance for Manufacturing Systems With Partially Ordered Process Plans.Robotics and Automation, 2001,17(6):819-827P
- [103] S.Venkatesh, Jeffrey.S.Smith. A graph-theoretic, linear-time scheme to detect and resolve deadlocks in flexible manufacturing cells. Journal of Manufacturing Systems,2003,22(03):220-238P
- [104] Hesuan Hu, Zhiwu Li. Local and global deadlock prevention policies for resource allocation systems using partially generated reachability graphs. Computers & Industrial Engineering, 2009,57(04):1168-1181P
- [105] Ezpeleta J,Tricas F,Colom,J.M.A banker's solution for deadlock avoidance in FMS with flexible routing and multiresource states. Robot. Automat,2002,18(04): 621-625P
- [106] Tricas.F. Prevention and Avoidance in Sequential Resource Allocation Systems. Ph.D. thesis 2003, Universidad de Zaragoza, Spain
- [107] MOHAN S, YALCIN A, KHATOR S K et al. Controller design and performance evaluation for deadlock Avoidance in automated flexible manufacturing cells. Robotics and Computer-Integrated Manufacturing,2004,(20):541-551P
- [108] J.Zajac. A.Deadlock Handling Method for Automated Manufacturing Systems. Manufacturing Technology,2004,53(01):367-370P
- [109] M.Lawley. Deadlock avoidance for production systems with flexible routing. Robot. Automat,1999,(15):497-509P
- [110] M. Lawley, S. Reveliotis, P. Ferreira. The application and evaluation of banker's algorithm for deadlock-free buffer space allocation in flexible manufacturing systems. Int. J. Flexible Manufact. System,1998,10(01):73-100P
- [111] LAWLEY.M., REVELIOTIS.S. Deadlock avoidance for sequential resource allocation systems. International Journal of Flexible Manufacturing Systems,2001,(13):385-404P
- [112] Fanti M P, Maione B, Turchiano B. Event control for deadlock avoidance in

- production systems with multiple capacity resources. *Studies in Informatics and Control*,1998,7(04):343-364P
- [113] 刘宏,叶尚辉,吴凤高.自动制造系统中的死锁.江苏机械制造与自动化,1997,(04):4-8 页
- [114] Coffman E G, Elphick M J, Sho shani A. System deadlock. *ACM Computing Surveys*,1971,(03):67-78P
- [115] Yalcin A, Boucher T O. Deadlock avoidance in flexible manufacturing systems using finite automata. *Robotics and Automation*,2000,16(04):424-429P
- [116] M.Uzam.An optimal deadlock prevention policy for flexible manufacturing systems using Petri net models with resources and the theory of regions.*International. Journal of Advance and Manufacturing Technology*,2002,(19):192-208P
- [117] Y.S. Huang, M.D. Jeng, X.L. Xie, D.H. Chung. Siphon-based deadlock prevention policy for flexible manufacturing systems.*Systems Man*, 2006,36(06):1248-1256P
- [118] Huang Y, Jeng M, Xie X, Chung S. Deadlock prevention policy based on Petri nets and siphons. *International. Journal of Production Research*,2001,39(02):283-305P
- [119] Cho H, Kumaran T K, Wysk R A. Graph-theoretic deadlock detection and resolution for flexible manufacturing systems.*Robotics and Automation*,1995,11(03):413- 421P
- [120] S. Venkatesh, Jeffrey S. Smith. A graph-theoretic, linear-time scheme to detect and resolve deadlocks in flexible manufacturing cells. *Journal of Manufacturing Systems*,2003,22(03):220-238P
- [121] Kumaran T K, ChangW, Cho H, Wysk R A. A structured approach to deadlock detection avoidance and resolution in flexible manufacturing systems. *Int J Prod Res*, 1994,32(10):2361- 2379P
- [122] 李志武,贾建援,王安荣.复杂并行共享资源与系统死锁.中国机械工程, 2003,14(04):280-285P
- [123] Wu N Q, Zhou M C. Avoiding deadlock and reducing starvation and blocking in automated manufacturing systems.*Robotics and Automation*,2001,17(05):658 -669P
- [124] H. Zandong, G. Lee. Application of Petri nets for deadlock analysis and avoidance in manufacturing systems. *International Journal of Advance and Manufacturing Technology*, 2005,(25):735-742P
- [125] Abdallah, I. B., Elmaraghy. A Deadlock-free scheduling in flexible manufacturing systems using Petri net. *International,Journal of production research*,2002,40(12): 2733-2756P
- [126] FantiM P, Maione B, Mascolo S, Turchiano B. Event based feedback control for

- deadlock avoidance in flexible production systems. *Robotics Automation*,1997, (13):347-363P
- [127] 赵振宏,肖田元.柔性制造单元的无死锁监控方法.计算机集成制造系统 CIMS,2001,7(05):14-18 页
- [128] ANTONIO R M, YU Hongnian , KELLEHER G. Hybrid Heuristic Search for the Scheduling of Flexible Manufacturing Systems Using Petri Nets. *Robotics and Automation*,2002,18(02):240-244P
- [129] LI Haolin. Algorithm of FMS production scheduling based on genetic algorithm. *Chinese journal of mechanical engineering*, 2000,36(09):91-97P
- [130] WU Xiuli. Research on flexible job shop dynamic scheduling problem. *Journal of System Simulation*,2008,20(14):3828-3832P
- [131] HUANG Zhonghua, WU Zhiming. Deadlock-free scheduling method for automated manufacturing systems using genetic algorithm and petri nets:Proceedings 2004 IEEE International Conference on Robotics and Automation, 566-571P .2004[C]
- [132] SIEMIATKOWSKI M, PRZYBYLSKI W. Simulation studies of process flow with in-line part inspection in machining cells. *Journal of Materials Processing Technology*,2006,171(01):27-34P
- [133] Z.X. Guo, W.K. Wong, S.Y.S. Leung, J.T. Fan, S.F. Chan. Mathematical model and genetic optimization for the job shop scheduling problem in a mixed-and multi-product assembly environment: A case study based on the apparel industry. *Computers & Industrial Engineering*,2006,(50):202-219P
- [134] 余琦玮,赵亮,潘双夏.基于遗传算法的柔性作业车间调度优化.组合机床与自动化加工技术,2004,(04):32-34 页
- [135] 赵天奇,陈禹六,李培根.基于零件虚拟工序队列的 FMS 动态调度研究.中国机械工程,1999,(10):1367-1369 页

攻读博士学位期间发表的论文和取得的科研成果

- [1] ZHAO Dan, ZHANG Jiatai. Deadlock state and its avoidance method in flexible manufacturing system with tool shared resource: 2009 IEEE International Conference on Mechatronics Automation, ICMA 2009,2928-2933P .(EI:20101012747113)
- [2] ZHAO Dan, ZHANG Jiatai, LI Lingli. Research on modeling and simulation of scheduling with tool failure in FMS. Key Engineering Materials. 2010(419-420): 101-104P. (EI:20094612453206)
- [3] ZHAO Dan, LIU Shaogang, SHU Haisheng. Research on tool flow deadlock based on Petri net and Graph theory. Key Engineering Materials.2011(450): 59-62P. (EI:20110113545231)
- [4] 赵丹, 舒海生, 张家泰, 赵刚. 基于工步备刀的 FMS 刀具流死锁的研究.哈尔滨工程大学学报,2009,(8):930-934 页. (EI:20094512433763)
- [5] 赵丹, 张家泰, 舒海生, 赵刚. 基于双重遗传算法的工件流与刀具流综合调度优化.西南交通大学学报,2010,(6):926-931 页. (EI:20110313600593)

致 谢

时光飞逝，转眼间五年多的博士求学历程即将结束。回首过去的几年，博士生活不仅丰富了我的知识，而且磨练了我的意志，令我受益匪浅，终生难忘。然而，这些都离不开众多老师、同学、朋友及亲人的关心和帮助，在此向他们致以最诚挚的谢意。

首先衷心感谢悉心培养与指导我的导师张家泰教授。感谢老师在论文完成过程中给予的精心指导和悉心关怀。恩师严谨的治学态度、高尚的敬业精神令我敬佩；渊博的知识、开阔的视野和敏锐的思维令我终生受益。正是由于导师的人格魅力与知识层次，使我在硕士研究生毕业后，又再次投到导师门下，在导师的指导下取得了今天的成果。

特别感谢舒海生副教授在系统建模和优化算法研究等方面给予我的启发性帮助；感谢赵刚教授、朱世范教授、刘少刚教授、孔凡凯教授、郑金兴副教授和应丽霞副教授对我论文工作的支持和帮助；感谢陈其廉教授、研究生姜斌和王士成在英文文献整理、翻译方面所做的帮助工作。

感谢我的同窗好友吴艳红博士、李玲莉博士和谷云庆博士几年来对我生活和学习上的关心和帮助。

向多年来一直鼓励本人不断进取的所有亲友们致以深切的谢意，谢谢你们对我的理解和支持。特别感谢我的妈妈，是你的勇敢和坚强给了我完成学业的信心和动力。

最后，再次向关心和帮助过我的老师、朋友、亲人和同学们表示衷心的感谢！

个人简历

赵丹，女，1978年2月出生于黑龙江省哈尔滨市呼兰区。

1997年9月至2001年6月，东北农业大学机械制造及其自动化专业学习，2001年6月获工学学士学位。

2001年9月至2004年3月，哈尔滨工程大学机械设计及理论专业学习，2004年3月获工学硕士学位。

2004年3月至今，哈尔滨工程大学任教。

2005年9月至今，哈尔滨工程大学机电工程学院攻读机械设计及理论专业博士学位。