

西安理工大学

硕士学位论文

面向Seru生产的生产单元构建与生产任务分配决策研究

姓名：廉洁

申请学位级别：硕士

专业：管理科学与工程

指导教师：刘晨光

201203

论文题目：面向 Seru 生产的生产单元构建与生产任务分配决策研究

学科专业：管理科学与工程

研究生：廉洁

签名：廉洁

指导教师：刘晨光 教授

签名：刘晨光

摘 要

Seru 生产是一种适用于多品种、小批量市场需求，兼具效率与柔性的生产方式。自产生以来，Seru 生产在日本制造业得到了迅速普及，并逐渐得到国际知名制造企业的认可。相较于企业实践，Seru 生产领域的学术研究还比较滞后，Seru 生产理论与方法体系远不完善，许多关键决策都尚未解决。

基于对 Seru 生产相关基础理论和国内外研究现状的分析，本论文研究了面向 Seru 生产的生产单元构建与生产任务分配两大关键决策。对于生产单元构建问题，本论文借助同质机器和可替换工艺路线来实现 Seru 生产下生产单元的生产完结性（kanketsu）。为使所研究问题更符合生产实际，本论文建立的数学模型综合考虑了加工时间、换装时间、加工顺序、产品需求量、机器加工能力、生产单元大小和生产单元数量等多个生产要素。根据数学模型的特征，本论文开发了一个基于非支配排序遗传算法-II 的混合遗传算法，并通过算例证实了数学模型的正确性和算法的有效性。对于生产任务分配问题，本论文结合 Seru 生产的本质特征，构建了符合 Seru 生产特征的生产任务分配决策模型。为使生产任务得到合理分配，生产能力得到充分利用，模型中从四个维度定义生产单元和产品，这四个维度分别是操作技能（部件种类）、技能熟练程度（品质等级）、装配时间（交付期）和加工成本（产品报价）。同时，本论文开发了一个启发式算法对模型进行求解，并通过与先前研究中的算法进行比对，证实该算法能够有效解决面向 Seru 生产的生产任务分配问题，并在最大完工时间、生产任务均衡性、生产能力利用率等多个方面具有良好表现。

本论文提出的生产单元构建和生产任务分配决策模型和求解算法，拓展了 Seru 生产领域的研究范围，为其他相关决策的研究奠定了坚实的理论基础，有助于完善 Seru 生产理论与方法体系。

关键词：Seru 生产；生产单元构建；生产任务分配；遗传算法；启发式算法

论文类型：应用基础研究

**Title: STUDY ON THE DECISION OF SERU FORMATION AND SERU
LOADING UNDER SERU SEISAN**

Major: Management science and engineering

Name: Jie Lian

Signature: Jie Lian

Supervisor: Prof. ChenGuang LIU

Signature: ChenGuang Liu

Abstract

Seru seisan is an advanced production mode, which merges both efficiency and flexibility. It is suitable to the demand characterized by high product variety and low production volume. Since its emerging, Seru seisan spread rapidly in Japanese manufacturing industry, and was gradually adopted by numerous well-known manufacturing giants around the world. Compared with its popularity in practices, research on Seru seisan starts just recently. There is a large room to improve the theory and methodology system of Seru seisan and a large number of key decision-making problems need to be solved.

On the basis of analysis of basic theories and literatures related to Seru seisan, this thesis solved Seru formation problem and Seru loading problem. For Seru formation problem, this thesis ensures the kanketsu of Seru by using identical machines and alternative processing routes. In order to make the problem more practical and realistic, many real life factors are simultaneously considered in the problem, such as processing time, setup time, operation sequence, production volume, machine capacity, cell size, quantity of cells etc. A hybrid genetic algorithm based on non-dominated sorting genetic algorithm-II (NSGA-II) is developed to solve the problem, and the computational results of numerical example validate the performance of the proposed algorithm. For Seru loading problem, a mathematical model is formulated for Seru loading problem considering features of Seru seisan. To make the Seru loading plan more reasonable and full use of production capacity, both Serus and products are defined from four dimensions, which are skill type/components, skill level/quality level, processing time /due date, and processing cost /expected price. To verify this model, a heuristic algorithm is developed. Then a numerical example is executed to test the computational performance of the proposed algorithm. The result is compared to that of a modified algorithm originally developed in previous research. The test results imply that the

proposed model is effective in solving Seru loading problem and it is superior to that modified algorithm with regard to make span, workload balance, and Seru capacity utilization.

The mathematical models and algorithms for Seru formation and Seru loading problems proposed in this thesis expand the research field of Seru seisan. They serve as the theory basis of research on other related decisions, and supplement the theory and method system of Seru seisan.

Key words: Seru seisan; Seru formation; Seru loading; Genetic algorithm; Heuristic algorithm

Type of thesis: Applied Foundation Research

Project Fund: National Natural Science Foundation of China (Grant: 71171161, 10801110)

1 绪论

1.1 研究背景与意义

随着全球化进程的逐步加快,制造企业间的竞争愈发激烈,其所面对的市场环境也发生了本质变化。首先,顾客需求呈现出明显的多样化趋势。随着生活水平的提高、消费意识的改变、以及可选择性的增加,顾客越来越难以满足于单一的、大众化的产品,追求多样化和个性化成为普遍的消费心理。其次,顾客定制需求逐渐增强。由于企业提供的标准化产品难以满足顾客的特殊需求,顾客可以参与到产品设计过程中,辅助企业设计出真正符合自身要求的产品。在需求变化的影响下,企业也逐渐由单一品种、大批量生产转向多品种、中小批量生产。此外,顾客对于产品质量、交付时间、更新换代速度等多方面的要求都有所提升。顾客的这种需求会一直持续并变得更加强烈,而满足这种需求需要一种具备成本优势的、能够快速制造、装配、运输中小批量定制产品的生产系统。适用于大批量生产的传送带流水线虽具备成本优势和较高效率,却缺乏应对定制产品市场变化的柔性和在小批量生产时发挥成本优势的能力,因此,在新的市场环境下暴露出各种弊端,进而产生了对新生产组织管理模式的迫切需求。大规模定制(Mass Customization, MC)是由美国未来学家 Alvin Toffler 在《Future Shock》中提出的对新生产方式的设想:以类似于标准化和大规模生产的成本和时间,提供客户特定需求的产品和服务^[1]。可见,实现大规模定制理念需要一种兼具效率与柔性的生产方式。

上世纪末,日本制造业遭遇了单一化需求市场的骤变,多品种小批量成为当时需求的主要特征,再加之当时日元大幅增值等多种因素的影响,特别是 1991 年泡沫经济的破灭,对日本制造业造成了重大冲击。面对严峻的挑战,日本一些制造企业展开了一场生产方式革新运动。1992 年产生于日本索尼株式会社的 Seru 生产就是一种能够满足多品种小批量市场需求、兼具效率与柔性的生产方式^[2]。Seru 生产是基于混流装配生产线的生产方式创新,其生产系统的基本元素——生产单元,是由混流装配生产线拆解转换而来的。原有的负责生产多种产品的混流装配生产线,拆解后形成多个生产单元,每个生产单元通常负责生产一种或少数几种产品,生产单元能够根据产品种类的变化随时进行重组或新建。这种生产组织形式不但保留了流水线生产的高效率和成本优势,还能够应对产品种类的变化和产品需求量的波动。与适用于大批量生产的传统装配流水线相比,Seru 生产的优势主要体现在柔性高、响应速度快、次品率低、在制品库存少、物料处理成本低、换装时间短、设备利用率高、计划与控制过程简化、节约生产空间、省人化、工人满意度高、责任感强等方面^[3,4]。

Seru 生产自产生以来在日本制造行业迅速传播,于 1997 年对日本制造企业进行的一次调查显示,有 23.2% 的受访企业已经采用了 Seru 生产,还有 25.0% 的受访企业正在计划实施^[5]。在电子行业,包括索尼、佳能、松下、富士通、日立、NEC 等在内的多家

著名电子产品制造企业都相继实施了 Seru 生产并取得了巨大成功^[6,7]。以佳能株式会社为例,1999 年至 2003 年间,该公司拆解了其下属 54 个工厂总长约 20000 米的装配流水线,并以 Seru 取而代之,这一举措节省了 720000 平方米的厂房空间,使在制品平均周转时间从 3 天缩短到 6 小时,总成本累计消减了 2 亿 3 千万日元,其子公司——佳能电子株式会社,在 1999 年至 2005 年间,利润从 11 亿日元增加到 118 亿日元,工厂空间节省了 70%,能源需求(水、电等)及二氧化碳排放量减少了 50%,平均生产率提高了 4 倍^[8]。佳能株式会社的董事长兼首席执行官 Fujio Mitarai 称, Seru 生产方式的实施使该公司成为一个高性能的组织^[9]。Seru 生产在日本制造业取得的巨大成功,使之在日本获得了“双 E (Ecology --- 环保, Economy --- 经济)”生产方式的美誉^[10]。

随着 Seru 生产在日本的应用与普及,其影响力也逐渐延伸到海外。目前,韩国三星电子、LG, 美国 Dell、Cmpaq 等国际知名制造企业也实施了 Seru 生产,并取得良好收益。与此同时, Seru 生产这一新兴的前沿领域内的学术研究也已经展开。目前,该领域的研究尚处于起步阶段,有关 Seru 生产的大量企业实践报道,大多出现在新闻、报纸及一些商业周刊,其实施详情的学术研究进程远远落后于企业实践。近几年只有为数不多的学术性研究论文出现在《工厂管理》、《IE 评论》等一些以介绍企业实践为主的杂志上,而在国际学术杂志上,这一领域的学术性研究论文更是屈指可数。同时,多年来学术界将 Seru 生产与传统单元化生产及精益生产相混淆,在一定程度上阻碍了该领域研究的发展。时至今日, Seru 生产系统设计、生产计划与控制等大量重要问题的解决理论与方法尚未建立, Seru 生产的理论与方法体系远不完善^[11]。

制造业是中国国民经济的重要支柱产业,中国制造业无与伦比的生产力也使中国被誉为“世界工厂”。然而,中国制造业还不够强大,仍有许多问题亟待解决。首先,中国的制造水平比较落后,大部分是基于廉价劳动力的劳动密集型生产;其次,研发能力欠缺,创新性差;第三,高级技术人才严重短缺,培训机制不完善。为了让中国成为真正的制造强国,必须从体制、政策、技术革新等多个方面入手进行改革,生产方式改革更是改变中国制造业现状的根本。面对严重的内部隐患和日益加剧的多品种小批量需求,中国亟须改进生产方式,建立一种适应中国生产实际的先进生产方式。近年来,中国学术界和企业界已经认识到 Seru 生产对中国制造业创新的重要意义,一些介绍 Seru 生产实施经验的学术研究成果和企业实施培训也相继出现,并表现出良好的发展势头。但是,中国学术界在该领域的研究还有待进一步深入,只有系统完善的理论知识体系才能很好地指导企业实践,成功引入 Seru 生产。

可见,深入研究 Seru 生产,既是立足国际前沿课题研究、推进科学发展的需要,也是中国企业创新生产方式、实现生存与发展的需要。本论文基于对大量相关文献的分析和对企业实施 Seru 生产经验的总结,深入研究了面向 Seru 生产的生产单元构建和生产任务分配决策,为设计 Seru 生产系统提供了理论依据和方法支持,因此,本论文对于面向 Seru 生产的实施决策分析与仿真研究具有重要理论价值和实践意义。

1.2 研究内容

本论文在系统研究 Seru 生产基础理论的基础上,着眼于解决面向 Seru 生产的生产单元构建与生产任务分配两大重要决策问题,深入探讨 Seru 生产下生产单元构建与生产任务分配问题的特征,分析问题解决思路并建立较为完善的方法体系,为中国企业借鉴和实施 Seru 生产,实现生产方式创新奠定理论基础。

论文首先研究面向 Seru 生产的生产单元构建决策。生产单元构建决策是 Seru 生产实施过程的第一步,也是 Seru 生产领域的四大主要研究方向之一。生产单元构建决策决定了企业的生产能力,对企业能否成功引入 Seru 生产,以及 Seru 生产系统能否正常运作都具有重要影响。生产单元构建问题主要解决生产系统中设备组和产品族的划分问题。该问题根据企业能够生产的产品种类,依据产品的几何特征或加工特征将所有设备划分成组,每一组设备构成一个生产单元。相应的,分配给同一生产单元的产品种类构成一个产品族,每个产品族包含一种或少数几种产品。合理的生产单元构建方案应根据市场需求和设备能力有效配置生产资源,在实现企业战略目标的同时达到资源利用率的最大化。

与传统的单元化生产方式不同,Seru 生产下的生产单元具备完结性特征,即对于每个生产单元而言,分配给该生产单元的产品的所有操作都能够在此生产单元内从头至尾全部完成,产品生产过程中无需使用其他生产单元中的资源。根据这一显著特征,本论文以具备完结性的、相互独立的生产单元为对象,以实现生产单元间工作量均衡为目标,研究了面向 Seru 生产的生产单元构建问题。通过对设备进行复制,将得到的同质机器分别分配到有需求的生产单元,以保证各生产单元包含所分配产品生产所需的所有设备类型,避免发生生产单元间物料移动,从而实现生产过程的完结性。为了更符合生产实际,所研究问题同时考虑了多项实际生产要素,如产品加工时间、可替换工艺路线、加工顺序、机器加工能力、产品需求量等。

以生产单元构建决策的研究结果为基础,论文进一步研究了面向 Seru 生产的生产任务分配问题。生产任务分配问题属于 Seru 生产领域四大主要研究方向之一的生产计划与控制范畴,主要解决如何将生产任务分配给已经建立好的生产单元,在满足产品加工需求的同时合理利用生产能力的问题。分配生产任务时,通常根据部件集合、品质等级、交付期、报价等属性,将产品划分为若干类,对于每类产品,至少有一个生产单元具备相应的生产能力。在给定的分配法则的指导下,综合考虑各类产品的加工需求和各生产单元的加工能力,最终得到最佳分配方案。生产任务分配方案直接影响着决定生产系统性能优劣的时间指标,如换装时间、总流程时间、完工时间等。合理的生产任务分配方案能够使生产流程顺畅平稳,大大减少延迟时间,保证订单按时交货。

Seru 生产系统的核心是经过岗位交叉培训,掌握多种技能的多能工。不同的多能工在操作技能种类和操作熟练程度上存在明显差异,操作熟练程度的差异通常体现在加工

时间和相应的加工成本上。由于 Seru 生产下的生产单元通常可以生产多个不同类型的产品，一类产品也可能具备多个可用生产单元，故产品类型和生产单元间时常出现多对多匹配现象。同时，由于多能工的存在，导致能够生产同类产品的多个可用生产单元间也存在较大的差异。为了提高生产任务和生产单元间的匹配度，本论文分别从操作技能（部件种类）、技能熟练程度（品质等级）、时间和成本四个维度来描述生产单元和生产任务的属性，以生产单元和生产任务各维度的属性值作为分配依据，以可变生产成本最小化为目标，分析满足最大完工时间最短原则的分配法则，构建面向 Seru 生产的生产任务分配方案。

1.3 研究思路与方法

本论文在深入分析 Seru 生产相关文献资料的基础上，对 Seru 生产的基础理论进行了科学地归纳和总结。同时，基于对日本企业实施 Seru 生产典型案例进行分析，对企业实施 Seru 生产的成功经验进行了理论化提炼，得到了对企业实践具有指导意义的实施框架和实施建议。上述分析深化了对 Seru 生产的理解与认识，为本论文研究奠定坚实的理论基础。在 Seru 生产基础理论和实施经验的基础上，通过系统对比 Seru 生产和传统单元化生产方式，提炼出面向 Seru 生产的生产单元构建和生产任务分配决策所具备的特征，进而展开针对两大重要决策的研究。

对于面向 Seru 生产的生产单元构建决策，首先对生产单元构建决策的国内外研究现状和发展趋势进行分析，然后根据 Seru 生产的本质特征和生产单元构建问题的基础理论，确定面向 Seru 生产的生产单元构建问题框架、影响因素、预期结果及创新点，并结合图示对所提出问题进行详细描述。基于所提出问题的特征，确定所研究问题的假设条件、目标函数和约束条件，并建立数学模型。接着，在对生产单元构建决策解决方法进行系统分析的基础上，结合所建数学模型的特征，开发基于非支配排序遗传算法（NSGA-II）的混合遗传算法对数学模型进行求解。最后，通过数学算例来验证所建数学模型的正确性和模型求解算法的有效性。

对于面向 Seru 生产的生产任务分配决策，首先对生产任务分配决策的国内外研究现状和发展趋势进行分析，结合 Seru 生产的本质特性，依据生产单元构建决策方案和生产任务分配基本流程，确定面向 Seru 生产的生产任务分配问题框架、基本要素、处理流程及创新点，并结合图示对所提出问题进行详细描述。基于所提出问题的创新点和处理流程，确定所研究问题的假设条件、目标函数和约束条件，并建立数学模型。结合所建数学模型的特征，开发并利用基于按需分解订单法则的启发式算法对数学模型进行求解。最后，分别用本文提出的启发式算法和文献中给出的算法对同一数学算例进行求解，通过对计算结果进行对比分析来证实所建数学模型的正确性和模型求解算法的良好性能。

1.4 研究框架

基于上述研究内容和思路，本论文的研究框架可概括为图 1-1。

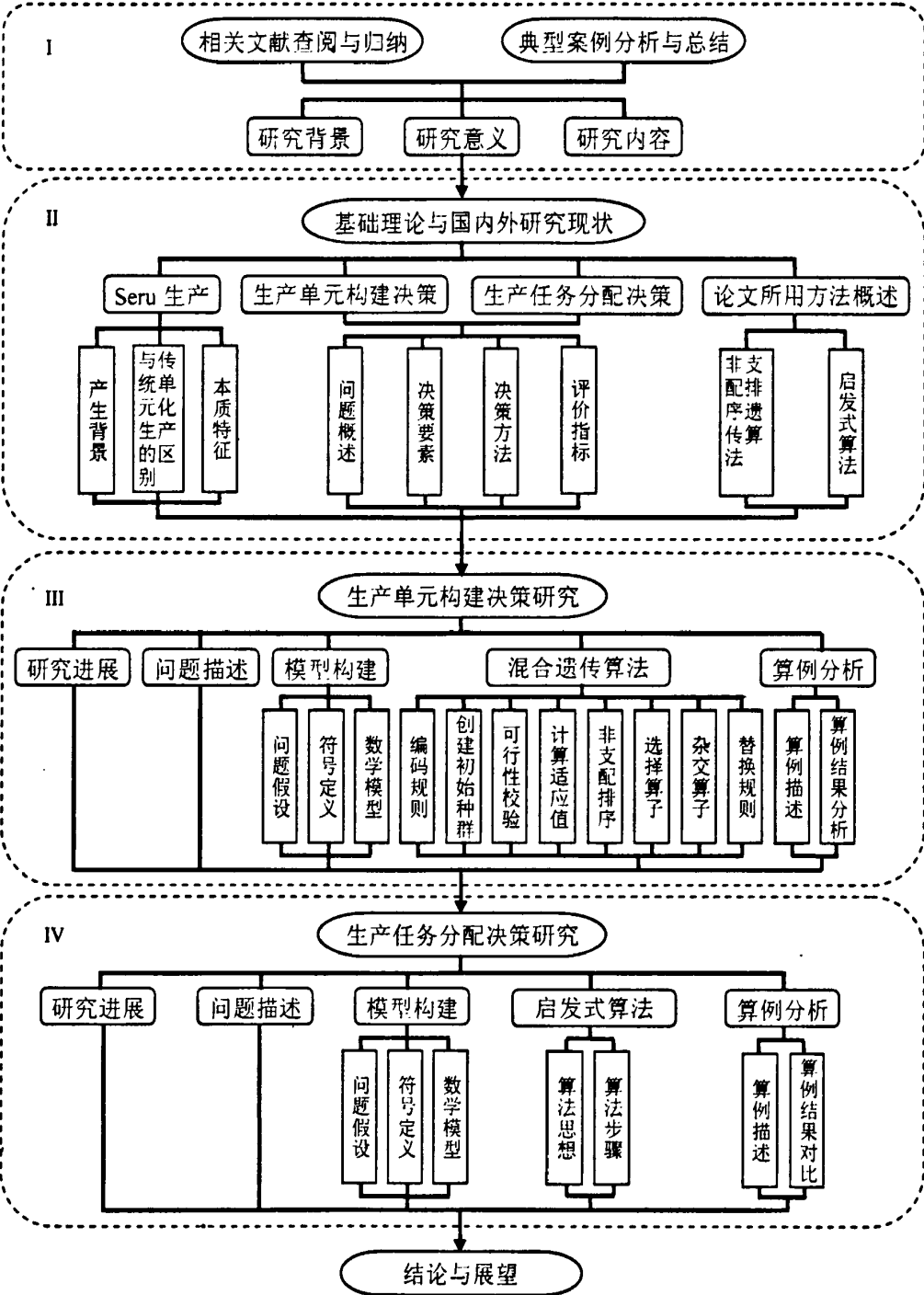


图 1-1 论文研究框架图

Fig. 1-1 The framework of the thesis

2 基础理论与国内外研究现状

2.1 Seru 生产

Seru 生产是在市场需求特征由单一品种、大批量向多品种、中小批量转变过程中催生的一种兼具效率与柔性的新型生产方式。Seru 生产是基于混流装配生产线的生产方式创新,它不但保留了混流装配生产线的效率,还具备能够应对频繁的产品种类变化和 product 需求量波动的柔性。自产生以来, Seru 生产在日本制造行业,尤其是电子行业,得到了广泛的应用并取得了良好收益,因此,它也被日本诸多行业誉为是“超越精益”的先进生产方式。如今,越来越多制造企业放弃了传统装配生产线,转而实施 Seru 生产。Seru 生产的普及呈现出良好发展势头,其应用范围正从日本本土企业向国际蔓延。然而,并不是所有实施 Seru 生产的企业都取得了成功,也有一些企业只得到了小幅改善,有的甚至不如从前,所以企业若想实施 Seru 生产并从中获益,必须充分了解其基础理论和运作机理,明确自身各方面因素是否适合实施 Seru 生产。本节从产生及其背景、与传统单元化生产的区别、本质特征等三个方面介绍 Seru 生产。

2.1.1 Seru 生产的产生背景

Seru 生产起源于日本著名的电子产品制造商——索尼株式会社。1992 年,在大野耐一(Ohno Taiichi)的学生、日本 PEC 产业教育中心长山田日登志(Yamada Hitoshi)的指导下,索尼公司美浓加茂工厂(现为索尼 EMCS 公司)将一条支持多种产品生产的混流装配生产线拆解为多条较短的专门用于生产一种产品的生产线^[7]。拆解出的短生产线和原来的长流水线具有相同的工作原理,只不过每一条短生产线只负责生产一种产品,这种拆解过程如图 2-1 所示。由于这种短生产线可以快速构造、优化和重组,在瞬息万变的市场中给索尼株式会社带来了很大益处。在持续改进中,为了更好地适应生产需要,这些拆分出的生产线不断缩短,在布局上也不断创新,其中 U 型布局最为普及。1994 年,索尼株式会社生产革新综合部部长金辰吉(Tatsuyoshi Kon)首次将这一拆解后的生产组织管理模式命名为“Seru Seisan”^[12]。Seru Seisan 意为单元化生产, Seru 是英文单词“Cell”的日语发音,而 Seisan 是汉语“生产”的日语发音,本文称之为 Seru 生产。

Seru 生产是日本制造业生产管理创新的时代产物,是日本国内外诸多因素综合作用的结果,其产生背景主要有以下几个方面:

(1) 多品种小批量的市场需求

随着经济的发展、生活水平的提高、及可选择性的增加,顾客需求日益呈现出多样化、个性化的特征。同时,各种高端技术尤其是 IT 技术的发展,导致产品生命周期大幅缩短,大大加快了产品更新换代速度。此外,产品品种的多样化直接导致每种产品需求量大幅降低,并呈现出剧烈的波动性。随着这种变化趋势的日渐明显,原有的以少品种大批量方式组织生产的企业无法适应市场的快速变化,从而引发了对高柔性生产组织管理模式的迫切

需求。

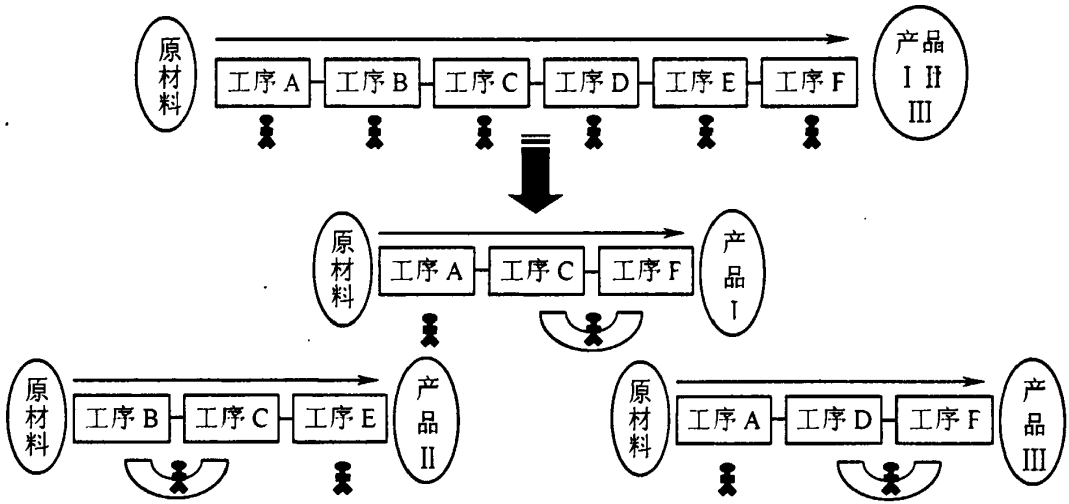


图 2-1 混流装配生产线拆解示意图

Fig. 2-1 The schematic diagram of dismantling of mixed-model assembly line

（2）缺乏柔性的流水线生产

在单一品种、大批量市场需求下，流水线生产的优点得到了充分发挥，例如较高的生产效率、标准化操作带来的高质量、规模经济及雇佣较低技能工人带来的成本效益。而流水线生产最大的不足就是缺乏柔性，因为针对一种或少数几种特定产品设计的传统流水线不但不能快速变化产品种类，还会时常出现生产不足或过量问题，而且这种缺陷随着市场需求多品种小批量化的加剧变得愈发明显。如今，柔性成为衡量生产系统优劣的重要指标，日本制造业管理人员将拆分流水线视为寻求生产柔性的一个重要途径^[13]。

（3）长期低迷萧条的经济环境

上世纪 80 年代至 90 年代初，日本电子行业诸多著名公司凭借着投入巨资的高度自动化生产系统，以无与伦比的竞争力成功地在国际市场上占据了主导地位。1990 年 1 月，伴随着股市、地价急剧下跌，日本经济泡沫彻底破灭，经济自此陷入长期低迷状态，进入了史称“失去的十年”经济萧条时期。受此影响，日本制造企业的自动化扩张趋势在上个世纪 90 年代初被遏制住。与此同时，消费需求的不稳定也使管理人员逐渐认识到高成本的自动化生产系统不一定能给企业带来预期的收益，因此，低成本、高效率的工厂自动化理念开始受到关注。

（4）竞争日趋激烈的国际市场

自 20 世纪 90 年代以来，中国源源不断的廉价劳动力供应使许多销往全球的产品价格一路走低，中国生产的低成本、高品质的产品，尤其是电子产品，对日本产业影响很大，导致日本产品的市场份额明显下降。另外，上世纪 90 年代日元大幅增值，从 1990 年的 145 日元兑 1 美元，到 1995 年 4 月出现了空前的最高纪录，日元对美元汇率为 79.75 日元兑 1 美元，这也使日本承受的外部成本压力骤然增大，严重影响了日本制造业的发展^[14]。

（5）日渐低落的员工士气

受日本传统文化的影响,日本企业大都采用年功序列制和终身雇佣制,这两种制度极易导致员工缺乏竞争意识和工作激情。加之传统流水线上分工过细和生产节拍的约束,使工人工作单调、紧张。如今,工人对提高个人技能和促进工作团队授权有着强烈的愿望,使企业更加重视工人培训,更加重视工作范围扩大化、工作丰富化和工作满意度的提升。

2.1.2 Seru 生产与传统单元化生产的区别

在学术文献中,Cell production 一词存在已久,它所代表的是盛行于欧美的单元化生产,我们称之为传统单元化生产。尽管 Seru seisan 与 Cell production 从字面意思上看没有区别,但是二者所指代的两种生产方式却存在本质区别,本小节从四个方面对这两种生产方式进行区分:

(1) 理论基础及形成方式

传统单元化生产由 Job shop 生产转化而来。在成组技术理论的指导下,其根据零部件或产品在外形和加工方法上的相似性,将零部件或产品划分为若干零部件族或产品族,并将原先按工艺相似性进行布局的设备转换为按零部件族或产品族组织生产,这样既保留了 Job shop 生产的柔性,又提高了生产效率^[15]。而 Seru 生产由装配流水线拆解转化而成,力求克服装配流水线刚性又尽力维持高生产效率^[13]。其理论基础至今仍是学术界争论的一个焦点,有学者指出, Seru 生产的理论基础是精益思想和索尼株式会社单人制生产理念的融合^[16, 17]。

(2) 应用领域和适用范围

传统单元化生产侧重在产品的加工生产阶段应用,已被成功应用于许多行业。针对美国 46 家金属制造企业的调查显示,每个生产单元基本涵盖成型、铸造、热处理、加工、装配、完成、检验和包装工序^[4]。而作为一种新兴的生产方式, Seru 生产主要被日本电子行业采用,且多用于装配生产阶段,属于装配型单元,一般仅包含装配、测试和包装工序^[18]。 Seru 生产主要用于依靠手工或简单设备生产的产品,对操作复杂、工序繁多的大型产品则具有一定的局限性。日本机械振兴协会经济研究所^[19]的调查数据与 Isa 和 Tsuru^[18]针对日本制造业的四个主要行业(日本通用机械,电气机械,运输设备和精密仪器)的研究指出,来自电气、机械及精密仪器行业的约 75% 的受访企业已经或计划实施 Seru 生产,然而运输设备行业中几乎一半的受访企业表示并没有实施的打算。

(3) 设备和工夹具

传统单元化生产的实施只是对设备按工艺布局的方式进行适当调整,但是 Seru 生产的实施,除了改变设备布局方式外,为了适应生产单元的频繁重组,传送带和大部分机器人被移去,重型设备被简化成轻巧、易移动的低成本设备。同时,传统单元化生产下的生产单元以包含多种产品(零部件)的产品(零部件)族为对象,而 Seru 生产下的生产单元仅致力于一种或少数几种产品的生产,相比之下, Seru 生产的设备及工具利用率更高。

(4) 系统柔性

两种生产方式都具有较高的生产柔性,但其背后的机理是完全不同的。从布局柔性来看,传统单元化生产广泛采用 U 型线布局。Seru 生产具有 I 型、U 型、L 型等多种生产单元布局,生产单元甚至可以组合出更为复杂的布局方式来应对产品需求的变化^[6]。从转产灵活性来看,传统单元化生产下的产品(零部件)族,一经确定,在一段时间内不会轻易改变。Seru 生产具备容易移动的工作台、轻型设备和持续培训的多能工,因此生产单元可以迅速重组,具有很高的应变能力。

尽管两种单元化生产存在诸多区别,但它们也具有一定的相似性,如都增加了对工夹具的投资、都降低了在制品和产成品库存、两种生产单元都具有自治特征等等。由于缺乏关于这两种生产方式区别与联系的系统性研究,导致 Seru 生产与传统单元化生产的混淆,从而在一定程度上制约了 Seru 生产在日本境外的广泛传播,特别是影响了国际学术界对这一生产方式的认识与研究。

2.1.3 Seru 生产的本质特征

与传统单元化生产和其他生产方式相比, Seru 生产具有以下几个突出的本质特征:

(1) Seru 生产的形成方式属于逆向转化。实施 Seru 生产的一个主要动因是为了克服装配流水线生产的刚性,以适应极具变化的多品种小批量市场需求。表现在形成方式上,就是由装配流水线拆解为若干生产单元。而基于成组技术的传统单元化生产,主要是为了提高生产效率,其表现形式为 Job shop 的重新组合^[15]。鉴于传统单元化生产形成较早,且国际上的研究成果丰硕,故可将其形成方式称为“正向转化”,与此对应, Seru 生产的形成方式则称为“逆向转化”^[20]。图 2-2 是两种单元化生产的形成方式示意图。

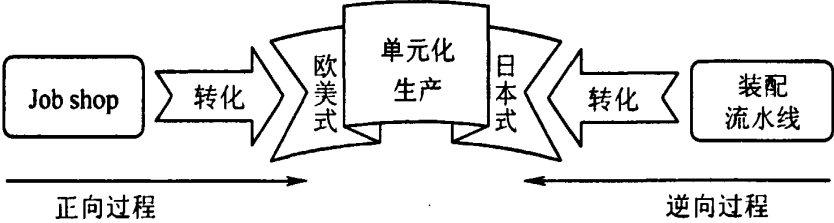


图 2-2 传统单元化生产与 Seru 生产形成方式示意图

Fig. 2-2 The schematic diagram of the generation of cellular manufacturing and Seru seisan

(2) 单个生产单元承担产品的大部分或全部生产过程。Seru 生产下,一个生产单元只致力于一种或少数几种产品的生产,对每一种具体的产品来说,其加工操作基本能够实现在一个生产单元内从头至尾全部完成,很少发生生产单元间物料移动。对于装配过程较复杂的大型产品,生产单元通常负责一部分相对完整的操作。为了使一种产品的大部分或全部生产操作(多是装配操作)能够在一个生产单元内完成,就要求所有必要的生产资源(如设备、工具、夹具)都安排在生产单元内部或临近的地方,操作人员也都必须是掌握多种技能的多能工。有别于装配流水线上操作时间均衡的单一任务,生产单元中工人的操作范围扩大,操作周期加长,每个工人负责不同数量的工序,而且每个工人负责的多道工序可以是非连续的^[21]。

(3) 生产单元是以人为中心的制造组织。多能工是实施 Seru 生产必不可少的关键因素,其他必要实施条件多是建立在工人完成向多能工转变的基础上,因此员工培训在整个 Seru 生产系统组建中占据十分重要的地位^[18]。多能工培训活动多在实施单元化生产之前就展开,培训内容不只局限于简单地让工人熟练其前导工序和后续工序,而要通过在同一生产单元内或多个生产单元间不断轮换工作来使工人能够掌握一种或几种产品生产所需的全部技能。轮换工作有助于增强工人对本职工作的责任感,因为他们会从中认识到自己的工作质量对其他工作和最终产品的影响。由于并非所有工人都能掌握全部技能,所以要在权衡工人能力的前提下,为不同的工人制定与之能力相匹配的培训计划,这样才能够最大程度地挖掘工人的潜力,提高工作满意度。经过培训的多能工可以在不同的生产单元里工作,因此可以通过适当调整工人分配情况来应对需求的变动。另外,在不同的生产单元里,可能有许多工人进行同样的操作,这样的工作分配方式也有利于促进工人间的技能交流,刺激创新。

(4) Seru 生产是倡导低自动化水平的生产方式。90 年代经济泡沫的破灭让大多数日本制造企业不得不放弃对高度自动化生产系统的追求,为了满足多个生产单元的需要同时达到削减费用的目的,大型装配流水线被拆解,一些不易复制、价格高昂的复杂设备被具有相同功能的构造简单、成本较低、易于构建和复制的设备所替代。最典型的例子是位于日本北部的斯坦利电气公司曾成功地将一个价值 3000 万日元的昂贵设备,重新设计成一个成本只有 20 万日元的简单设备,花费仅仅是原来的 1/150^[22]。

(5) 生产单元可以及时发现并修复产品缺陷。多能工能够操作的工序范围比较广,他们较容易发现前面工序中产生的缺陷,同时 U 型生产布局使工人一旦检查到产品的缺陷就可以立刻进行处理。与装配流水线不同,一个生产单元内的产品缺陷修复不会影响整个生产系统的正常运行,在遇到困难时,多能工可以互相帮助,有助于快速有效地解决问题。

(6) 实施 Seru 生产可以节省大量空间。传统流水线的长度取决于线上工人的间距和工人的数量,其直线布局方式需要足够的空间来实现。相比而言,生产单元则具有很多灵活的布局方式来适应不同企业的实际情况。在生产单元中,相邻工作台之间的距离和工人面对面的距离都明显缩短,这样不仅有利于工人之间的交流,也使那些轻型产品便于依靠人力移动,而体积较大的产品则可使用轻便的推车来移动,这样整个生产流程变得更加简捷高效。

(7) 生产单元具有多种布局方式。U 型、L 型、I 型都是常见的生产单元布局方式,通过生产单元的组合还能得到一些像花型、心型等较复杂的布局。当需求波动时,生产单元的形状和规模可以在几分钟内迅速调整,第一时间做出反应。生产单元布局的灵活性取决于多能工掌握多种技能的程度和简易设备的可移动性,尽管生产单元的布局多变,但都力求尽可能减少占用的资源,降低在制品库存。

(8) 生产单元是一种高度自治的组织。从横向来看,每个生产单元都相互独立,承

担具有一定完整性的操作流程,它通过出入口与系统其他部分相连;从纵向来看,为了使生产过程更加顺畅,生产单元具有监督和控制生产过程的管理职能。然而,生产单元的自治并不意味着它总是独立于其他部分,也不能理解为它是不受控制的,无论生产单元具有多大的管理职能,总会有对其进行监督控制的部门。

(9) 生产单元是持续优化的学习型组织。工人会主动学习先进知识,提高操作技能,培养应对意外状况的能力,他们与管理层有着良好的沟通,敢于提出自己的意见和见解,以促进生产过程的不断优化和完善。管理层也基于生产单元的完善、进化水平,提出合理的管理目标,促使生产单元更好地发展。

2.2 生产单元构建问题

2.2.1 生产单元构建问题概述

生产单元构建决策主要包括两大问题:产品(零部件)分类和机器分组。产品(零部件)分类是根据产品(零部件)的设计或工艺特征对其进行编码,然后通过计算相似性系数确定产品(零部件)间的相似程度,进而将相似度最高的一类产品(零部件)划归为一个产品(零部件)族。机器分组是依据机器的加工能力,将能够生产一个产品(零部件)族或一类产品(零部件)的机器组合成生产单元。通过对产品(零部件)和生产资源进行分组,能够显著提高生产效率,同时减少生产成本。

根据生产单元设计时所侧重的对象的不同,可以得到三种生产单元构建策略。若以产品(零部件)族为导向,则首先根据产品(零部件)的设计或加工特征划分产品(零部件)族,然后根据所建立的产品(零部件)族来建立相应的生产单元;若以机器为导向,则先根据机器加工能力将机器分组建立生产单元,而后根据产品(零部件)工艺路线的相似性将产品(零部件)分配给各生产单元;若对产品(零部件)和机器的侧重相同,则同时建立生产单元和产品(零部件)族。

生产单元构建问题通常表示为将二进制矩阵转换为分块对角矩阵的问题。用于解决生产单元构建问题的许多方法中都使用了机器-零件关联矩阵 $A=[a_{ij}]$, 其中 $a_{ij}=1$ 的元素表示零件 j 需要在机器 m 上加工。机器-零件关联矩阵根据产品的工艺路线卡构建,初始矩阵中机器和零件之间的关系并不明显,通过使用各种方法调整机器和零件的位置,最终可以形成分块对角矩阵,其中每一块所对应的机器和零件分别构成一个生产单元和零件族。图 2-3 展示了机器-零件关联矩阵转换前后的区别,转换后的矩阵将元素分为两大块,分别代表两个生产单元。图中机器 2,4,5 组成一个生产单元,机器 1,3 组成另一个生产单元,所对应的零件族分别为 $\{3,6,5\}$ 和 $\{4,1,2\}$ 。区域 A 中值为 0 的元素代表空闲机器,区域 B 中值为 1 的元素是例外元素,其生产过程中有一些工序需要在其他生产单元完成。

大部分文献在解决生产单元构建问题时都是在减少生产单元间物料处理次数或成本的基础上建立尽可能小的、独立的生产单元,因为这样的生产单元有利于更好地体现单元化生产的优势。Selim 等人^[23]指出,“完美的”生产单元是相互独立的,在生产过程中不

会发生生产单元间物料移动。也就是说，产品的生产过程具备完结性（Kanketsu），产品的所有工序能够从头至尾在一个生产单元内完成，无需借助其他生产单元。具备完结性的生产单元虽然能够减少生产单元间物料处理费用，但是也会带来大量的新机器购买费用。现有文献在解决生产单元构建问题时，都或多或少地存在生产单元间物料移动，只有少数研究能够彻底消除生产单元间物料移动，建立真正相互独立的生产单元。

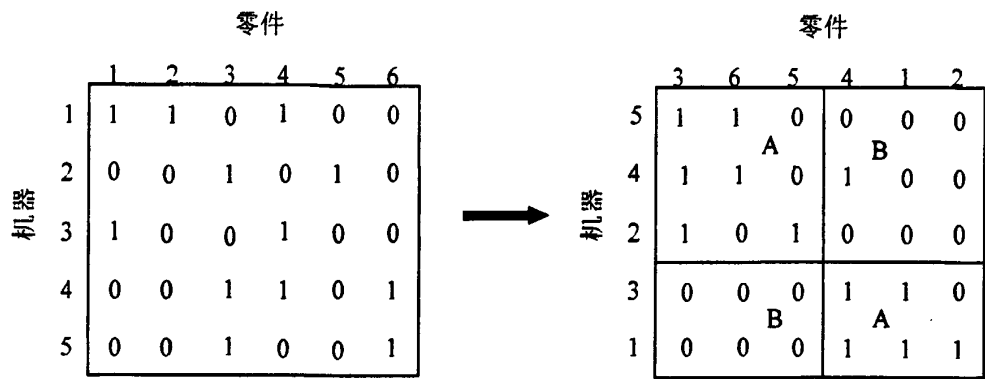


图 2-3 机器-零件关联矩阵转化示意图

Fig. 2-3 The schematic diagram of the transforming of machine-part incidence matrix

解决生产单元构建问题时经常使用的目标函数除了物料移动外，还包括相似性系数、机器投资、加工成本和工作量均衡。相似性系数出现最早，由于早期的研究考虑因素较少，多是借助排列算法对机器-零件关联矩阵的行和列进行交换得到分块矩阵来确定生产单元和产品族，因此，用相似性系数最大化作为目标函数能够很好地将相似元素归类。同时，由于相似性系数的计算方法成熟，且易于计算，被早期的大量文献采用。随着所研究问题的复杂化，成本、时间等指标也常被作为衡量生产单元构建方案的标准。对于考虑成本因素的研究，除了物料处理成本，最常考虑的因素还包括机器投资费用、产品加工成本和单元重组成本。重视生产系统平衡性和稳定性的研究则经常考虑工作量均衡程度。此外，工作量均衡也可作为约束条件，这样可以同时实现工作量均衡和其他目标。

生产单元构建问题大致分为静态生产单元构建，动态生产单元构建和虚拟生产单元构建三种。静态生产单元构建问题针对一个给定时期内的静态产品需求来建立生产单元。该问题假设在构建生产单元时，所针对的时期内的产品需求全部已知，且产品种类和需求数量不会发生变化。所得到的生产单元和产品（零部件）族构建方案一经给定，不会发生变化，且只对给定时期内的产品需求有效。此类问题适用于产品需求在一定时期内相对稳定的环境。如果产品需求多变，静态生产单元构建问题得到的方案很快就会失效。动态生产单元构建问题延长了静态生产单元构建问题的时间轴，考虑多个时期下的生产单元构建问题。多个时期下产品需求种类和数量均会发生变化，这些变化可能是随机的，也可能是能够预测的。由于产品需求的变化，前一个时期下的最优方案有可能不是下一个时期下的最优方案，因此，该问题的难点就是在考虑不同方案间转换成本的条件下如何选择各个时期下的方案才能达到整体最优。以上两类问题均会按照生产单元构建方案来布局设备，而虚

拟生产单元构建问题不会改变生产设备的位置,只是在逻辑上暂时将处于不同位置的设备视为一个生产单元,用来加工所对应的产品(零部件)族,因此,这种问题适用于产品需求不断变化、且不可预测的环境。当产品需求发生变化,可以随时打破虚拟生产单元内机器间的联系,并组建新的生产单元,这样可以避免生产设备的频繁移动。

2.2.2 决策要素

生产单元构建问题是非常复杂的问题,要考虑很多生产要素。Defersha 和 Chen^[24]总结了生产单元构建问题需要考虑的 21 种要素,并通过对近期发表的 19 篇文章进行分析,得出了各要素的出现频次。表 2-1 给出了这 21 种要素,其中加下划线的要素是现有文献中最常考虑的要素。受到问题复杂度的影响,在同一个问题中无法同时考虑所有要素,现有文献中提到的许多方法只能同时考虑若干项,本节介绍与本论文相关的四项要素。

表 2-1 单元化生产系统要素表

Table 2-1 The list of manufacturing attributes of cellular production

1.可替换工艺路线	(a) <u>选择最佳路线</u>	11.生产单元/零件族规模约束	(a) <u>生产单元大小</u>
	(b) 多条路线共存		(b) 零件族大小
2.需求波动情况	(a) <u>确定性需求</u>	12.零件移动次数 (物料处理成本)	(a) <u>生产单元间移动</u>
	(b) 可预测需求		(b) 生产单元内移动
3.动态生产单元重组		13.设备布局	(a) 生产单元间布局
4.工作量均衡	(a) 生产单元间		(b) 生产单元内布局
	(b) 生产单元内	14.工人分配	
5.批量分割		<u>15. 机器能力</u>	
6.零件加工需要的工具类型		<u>16. 同质机器</u>	(a) 单个生产单元内
7.机器上可用的工具类型			(b) 整个生产系统
8.机器邻近约束	(a) 相互独立	<u>17.机器投资成本</u>	
	(b) 允许组合	18.外包成本	
9.加工顺序	(a) <u>确定物料流等级</u>	19.工具消耗成本	
	(b) 衡量零件相似性	<u>20.单位加工时间</u>	
10.换装成本/时间	(a) 换装成本	21.加工成本	
	(b) 换装时间		

(1) 可替换工艺路线

可替换工艺路线通常出现在离散的、多批次、小批量的生产环境中,一种产品若存在多条可选择的工艺路线,则说明其加工路线柔性较高,因此也丰富了生产单元构建方案。当一道工序可以在多台机器上加工时,这些机器就可以被视为是备选工艺路线。大多数研究不考虑可替换工艺路线,这有可能会增加加工成本和新机器购买成本。若例外元素存在可替换工艺路线,就可以通过调整工艺路线来减少生产单元间物料移动。对于考虑多条工艺路线的生产单元构建问题,也只是在最终解决方案中为每个产品(零部件)选择一条最

佳的工艺路线,考虑同时选择多个可替换工艺路线的研究成果屈指可数。Defersha 和 Chen^[24]在解决生产单元构建问题时考虑了可替换工艺路线,与以往的研究不同,他们的问题中允许多条可替换工艺路线共存并在具备成本优势的情况下共同分担产品的生产任务。Chan 等人^[25]提出了一个同时解决生产单元构建问题和生产单元布局问题的两阶段方法,可替换工艺路线是其第一阶段考虑的主要因素。

(2) 工作量均衡

单元化生产系统中工作量均衡分两种情况,一种是生产单元内机器或工作台之间的工作量均衡,另一种是生产单元间各生产单元的工作量均衡。工作量均衡有利于生产系统的平稳运行,能够有效减少在制品库存、最大完工时间、总流程时间和延迟时间。若单元化生产系统中工作量分配不均衡,就会出现一些生产单元过载,而一些生产单元过于空闲的情况。Kim 等人^[26]研究了一个多目标生产单元构建问题,问题以减少生产单元间零件移动次数和机器间工作量不均衡为目标函数,同时解决了机器分组、产品分类和产品工艺路线选择问题。Al Kattan^[27]在研究流水车间环境下的生产单元构建问题时,提出了一个实现各工作台和机器间工作量均衡的有效方法。

(3) 零件移动次数(物料处理成本)

几乎所有的生产单元构建决策解决方法都致力于如何减少物料处理成本,尤其是生产单元间物料移动产生的生产单元间物料处理成本。

① 生产单元间物料移动

生产单元和产品族建立后,一些例外元素(Exceptional element)的某些工序无法在它所在的生产单元内加工,要借助其他能操作该工序的生产单元来完成生产,此时就会发生生产单元间物料移动。生产单元间移动会直接破坏生产单元的独立性,增加生产成本,延长生产流程时间,降低资源利用率和生产效率,但是,生产单元间移动却可以使一些未被有效利用的机器得到充分利用。从移动次数和距离上来看,生产单元间物料移动成本要比生产单元内物料移动成本高得多,因此,早期的研究中大部分问题只考虑到了生产单元间物料移动成本。Albadawi 等人^[28]2005 年用一个以例外元素最小化为目标函数的两阶段数学模型实现了依次建立生产单元和产品族的问题,该模型间接地减少了生产单元间物料移动。2006 年,Chan 等人^[25]用一个两阶段方法同时解决了生产单元构建问题和生产单元布局问题,除了建立产品族和生产单元外,该方法还分析了加工顺序对减少生产单元间移动距离的影响。

② 生产单元内物料移动

生产单元内物料移动是由生产单元内的空闲机器造成的,这些机器上没有分配生产任务,从而打断了生产过程的连续性,迫使产品在生产过程中在不同机器间移动。与生产单元间物料移动相比,生产单元内物料移动耗费的时间和费用相对较小,没有对系统性能造成直接的、重大的影响,因此,一直都被忽略不计。近年来,随着研究的深入,越来越多研究者注意到生产单元内物料移动除了会产生生产单元内物料移动成本外,还会间接降低

机器利用率。现在,许多研究将生产单元内和生产单元间物料移动合并成了一个系统性能评价指标。基于一系列启发式法则,Vitanov 等人^[29]提出了一个用来解决生产实践中生产单元构建问题的决策支持工具,问题中同时考虑了生产单元内和生产单元间物料移动。以降低总物料移动成本(包括生产单元内和生产单元间)和减少例外元素为目标,Wu 等人^[30]开发了一个分层遗传算法来同时确定生产单元构建和布局方案。

(4) 同质机器

同质机器是指属于同一种机器类型,加工能力完全一致的一类机器。同质机器可以通过购买和机器复制来获取。在生产单元构建问题中,通过将同质机器分配到所有有相应加工需求的生产单元中,可以彻底杜绝生产单元间物料移动,保证生产单元的独立性和产品加工过程的完结性。虽然同质机器能提高生产系统性能,但同质机器的数量越多,所产生的购买或复制成本越高,也更容易出现部分机器利用率过低的现象。尽管从长期考虑,生产过程中减少的生产单元间物料移动成本可能抵消机器购买或复制成本,但大多数企业不会轻易选择投资新机器,除非生产能力严重不足。若要购买或复制机器,在确定同质机器的最佳数量时必须权衡企业预算和实际生产需求。Lee 和 Chen^[31]将复制得到的同质机器间的工作量均衡作为评价系统性能的重要指标,利用一个加权三阶段方法来建立生产单元和产品族。在分析零件加工需求和机器加工能力的基础上,Defersha 和 Chen^[24]提出了一个数学模型来解决生产单元构建问题,该模型结合了多个生产要素,如可替换工艺路线、批量分割、加工顺序和同质机器。

2.2.3 决策方法

生产单元构建问题是一个组合优化 NP 困难问题。生产单元构建问题的解决方法大致可以分为六大类,即聚类分析法,数学规划法,启发式算法,元启发式算法,人工智能方法和混合元启发式算法^[32]。

聚类分析法通过将不同个体聚合成组,使得到的每个分组内的元素与组内其他元素之间具有较高的关联,而与其他组内的元素之间具有较低的联系。聚类分析方法包括基于阵列的聚类、分层聚类和无分层聚类。基于阵列的聚类通过适当地调整机器-零件关联矩阵中行和列的顺序得到分块对角矩阵,从而确定生产单元和产品(零部件)族。分层聚类包括两个阶段,首先,找出机器和产品(零部件)间的相似或不相似的地方,然后,机器和产品(零部件)被分割为若干较大的生产单元,接着,每个生产单元再被分割为若干较小的组,分割出的较小的组继续被划分,直到出现无法被分割的组为止。分层聚类的过程分为两类,一类是先将所有数据放进一个大组,然后不断分裂,直至每个机器(零部件)都在一个独立小组的分裂方法;另一类是从独立的小组入手,将不同小组组合成大的部分的凝聚方法。无分层聚类是一种借助相似性指标的迭代方法,这种方法需要提前确定类的数量。聚类分析方法出现最早,它操作简单,且花费时间较短,广泛应用于将机器-零件关联矩阵转化为分块对角矩阵。但是通常只能考虑一个目标函数(如生产单元间物料移动最

小化)。聚类分析法中使用的机器-零件关联矩阵只包含机器的加工技能和产品工序,许多实际生产要素都没有考虑,因此,这种方法得到的生产单元构建方案实用性较低。

生产单元构建问题下建立的数学模型通常都是非线性或线性整数规划问题。数学规划法被广泛应用于各种问题环境中,它的显著优势就是能够综合考虑多种生产要素,如加工顺序、可替换工艺路线、换装和加工时间、以及同质机器等。但是,考虑的因素越多,解决模型需要的数据就越多,计算复杂度也越大,因此,这种方法不利于解决数据规模较大的问题。Won 和 Lee^[33]提出了一个经过修改的 p -中值方法来解决生产单元构建问题,问题的目标是实现同一个生产单元内机器间的相似性系数之和最大。修改后的 p -中值方法克服了原始 p -中值方法的缺点,易于实现且不受所解决问题大小的限制。

考虑到生产单元构建问题的 NP 困难和组合优化特征,启发式算法、元启发式算法和混合启发式算法被用于在合理时间内解决具备全局最优解或近似全局最优解的生产单元构建问题^[32]。启发式算法在实际中得到广泛的应用,相对于那些严格要求的约束条件,启发式算法能够提供可选择的框架来解决问题。尽管启发式算法不能保证得到最优解(通常只能得到次优解),但它能在合理的时间内得到人们可以接受的解。Chan 等人^[34]开发了一个启发式算法用于解决只考虑生产单元内物料流的机器分配问题。该算法使用一种自适应方法通过查看两个机器间合并零件流的权重将机器与生产单元关联起来。该问题的目标函数是降低每个生产单元内的总移动次数,即总移动距离。Kim 等人^[35]研究了一个更加复杂的多目标生产单元构建问题,该问题同时实现生产单元间零件移动次数总和最小化和机器工作量均衡最大化。这个问题用一个两阶段启发式算法解决,第一阶段确定零件的代表工艺路线和工艺路线族,第二阶段将剩余的零件工艺路线分配给工艺路线族。

与启发式算法相比,元启发式算法的通用性较强。在过去的三十年中,元启发式算法被用于解决大多数 NP-困难组合优化问题。生产单元构建问题属于这类问题,不但具备较高的复杂性,且不易于找到最优解。元启发式算法包含两部分,即轨迹方法(模拟退火和禁忌搜索)和基于种群的方法(遗传算法、粒子群优化算法、分散搜索算法和蚁群算法)。

模拟退火算法和禁忌搜索算法的共同点是它们的搜索程序始于一个初始状态(初始解),在解空间内会形成搜索轨迹。模拟退火算法是元启发式算法中出现最早的算法,它是一种通用概率算法,用来在固定时间内寻求在一个大的搜寻空间内找到最优解。Souilah^[36]给出了通用模拟退火算法,并展示了如何用这种算法将机器划分为生产单元、设计生产单元内布局和各生产单元在工厂内的布局。禁忌搜索算法是应用于组合优化问题的最成功的元启发式算法之一,它是一种全局性邻域搜索算法,模拟人类具有记忆功能的寻优特征。它通过局部邻域搜索机制和相应的禁忌准则来避免迂回搜索,并通过破禁水平来释放一些被禁忌的优良状态,以最终实现全局优化。Wu 等人^[37]研究了考虑工艺路线、产品需求量和生产单元大小的生产单元构建问题。该问题旨在将整个制造车间分解为若干生产单元,从而实现生产单元间的总零件流最小。为了解决该问题,论文中给出了一个包含动态禁忌任期和长时记忆结构在内的禁忌搜索算法。分组-分配和随机方法是文中使用的两

种能够快速产生初始解的方法。

遗传算法、粒子群优化算法和分散搜索算法属于进化算法,进化算法和蚁群算法都是基于种群的搜索方法。进化算法模拟自然界生物优胜劣汰的生存法则寻求最优解,由于其搜索过程始于解空间内一个种群内的所有解而非一个单一的解,因此,进化算法得到了广泛应用。Solimanpur 等人^[38]建立了一个多目标整数规划模型来解决独立生产单元构建问题,并提出了一个具有多个适应值方程的遗传算法来求解模型。文中的遗传算法与先前的方法存在两点区别,它使用系统一致设计技术来确定搜索方向,然后从多个方向而非单一方向搜索解空间。Durán 等人^[39]提出了一个改进粒子群优化算法,该算法与标准粒子群优化算法不同的地方是不使用速率向量。该算法用于解决基于生产单元间物料移动次数最小化的生产单元构建问题,计算结果显示该算法几乎在所有算例中都找到了最优解。

用于生产单元构建问题的人工智能方法主要包括神经网络方法和模糊理论。神经网络具有稳健性和适应性,因此,其应用范围也相对广泛。Yang 和 Yang^[40]提出了一个改进自适应谐振理论神经网络学习算法,为了更加有效地选择可信的警戒值,该算法中的警戒参数是根据已知数据估计得到的。作者将这一算法应用于生产单元构建问题,并证明了该算法在解决生产单元构建问题时的有效性。为解决动态生产单元构建问题下的扩展混合整数规划模型,Safaei 等人^[41]开发了一个基于模糊规划的方法。该问题中将每个计划期零件需求的波动和可用生产能力视为模糊数,并作为目标函数方程和机器-零件关联矩阵的系数,借此确定在给定约束条件下能够满足模糊目标函数的最优方案。

混合元启发式算法在近几年兴起,它通过将元启发式算法与其他最优化技术相结合来构造出更加有效、柔性更高的方法。Nsakanda 等人^[42]研究了考虑多工艺路线的生产单元构建问题,并提出了一个基于遗传算法和大规模最优化技术的混合元启发式算法。James 等人^[43]将标准分组遗传算法和局部搜索方法相结合,用来解决生产单元构建问题。与标准分组遗传算法相比,这种混合算法能得到更优的解,大大降低了解的差异。

2.2.4 评价指标

在过去的研究中,生产单元构建问题取得了丰硕的研究成果,为了有效对比不同解法下生产单元构建方案的优劣,文献中给出了许多评价系统性能的定量指标,本节简要介绍四种常用的评价指标。

(1) 分组效率 (Grouping efficiency)

分组效率由 Chandrasekaran 和 Rajagopalan^[44]于 1986 年提出,该指标是最早提出的用来评价生产单元构建方案的指标之一。这个指标依据生产单元内的机器利用情况和生产单元间物料移动来判断解的优劣。分组效率是 η_1 和 η_2 两种效率的加权平均值,如式(2.1)所示:

$$\eta = \omega\eta_1 + (1-\omega)\eta_2 \quad (2.1)$$

其中

$$\eta_1 = \frac{o-e}{o-e+v} \quad (2.2)$$

$$\eta_2 = \frac{MP-o-v}{MP-o-v+e} \quad (2.3)$$

M 是机器数量, P 是零件数量, o 是矩阵中所有值为 1 的元素的数量, e 是区域 B 中值为 1 的元素的数量, v 是区域 A 中值为 0 的元素的数量。与图 2-3 对应, η_1 表示区域 A 中所有值为 1 的元素的数量与区域 A 中所有元素 (包括值为 1 和 0 的元素) 的数量的比值, η_2 表示区域 B 中值为 0 的元素的数量与区域 B 中所有元素 (包括值为 1 和 0 的元素) 的数量的比值。

通常情况下建议让权重 w 等于 0.5, 也可根据设计生产单元时对生产单元间物料移动和机器利用率的侧重灵活调整权重值。分组效率的取值在 0 到 1 之间, 越接近 1 时, 表示结果越好。但是, 分组效率对于解的辨别能力较差, 即便是那些具有大量例外元素的非常差的解, 其分组效率值也能达到 0.77^[45]。

(2) 分组效力 (Grouping efficacy)

分组效力由 Kumar 和 Chandrasekaran^[46]于 1990 年提出, 该指标不但克服了分组效率的缺陷, 能够很好地辨别解的好坏, 而且指标值不受问题大小的影响。分组效力如式(2.4)所示:

$$\tau = \frac{1-\psi}{1+\phi} = \frac{o-e}{o+v} \quad (2.4)$$

其中 o 是矩阵中所有值为 1 的元素的数量, e 是区域 B 中值为 1 的元素的数量, v 是区域 A 中值为 0 的元素的数量。与图 2-3 对应, ψ 表示区域 B 中值为 1 的元素的数量与矩阵中所有值为 1 的元素的数量的比值, ϕ 表示区域 A 中值为 0 的元素的数量与矩阵中所有值为 1 的元素的数量的比值。分组效率的取值也在 0 到 1 之间, 当取值为 1 时是最优解, 解中不包含例外元素和空闲设备。

(3) 键能指标 (Bond energy measure)

McCormick 等人^[47]利用键能指标将二进制矩阵转化为分块对角形式, 其核心思想是将矩阵中所有非零元素尽可能紧密地排列在一起。键能指标如式(2.5)所示:

$$\eta_{BE} = \frac{\sum_{m=1}^{M-1} \sum_{p=1}^P a_{mp} a_{(m+1)p} + \sum_{m=1}^M \sum_{p=1}^{P-1} a_{mp} a_{m(p+1)}}{\sum_{m=1}^M \sum_{p=1}^P a_{mp}} \quad (2.5)$$

键能用来衡量聚簇矩阵的相对丛化性 (the relative clumpiness), 因此, 它的值越接近 1 越好。

(4) 设备利用指数 (Machine utilization index)

设备利用指数是由 Miltenburg 和 Zhang^[48]提出的用来衡量生产单元内设备利用情况的指标。设备利用指数可表示为:

$$\eta_g = \eta_u - \eta_m \quad (2.6)$$

其中,

$$\eta_u = d/(d+v), \quad (2.7)$$

$$\eta_m = 1-(d/o), \quad (2.8)$$

d 是区域 A 中值为 1 的元素的数量, o 是矩阵中所有值为 1 的元素的数量, v 是区域 A 中值为 0 的元素的数量, η_u 衡量生产单元内设备的利用情况, η_m 衡量生产单元间零件移动次数。设备利用指数的取值范围从 -1 到 1, 取值越大越好。

2.3 生产任务分配问题

2.3.1 生产任务分配问题概述

生产任务分配问题属于生产计划与控制范畴, 生产计划与控制决策会直接影响整个生产系统的时间框架 (如不同产品批次的生产开始和结束时间), 进而影响生产流程的平稳流畅和整个生产系统的性能。在单元化生产产生之前, 大量研究都关注于 Job shop 下的生产计划与调度问题。从 20 世纪 50 年代开始, 此研究领域涌现出大量研究成果, 这些研究大致可以分为四大类: 订单投放、到期日分配、优先级分配和工人分配^[49]。Ahmed 和 Fisher^[50] 建立了 Job shop 仿真模型, 通过实验分析了这四类问题之间的联系。

Kanet^[51] 研究了 Job shop 下的订单分配问题, 他用工作量限制来控制订单分配过程, 该研究发现工作量限制会导致生产过程中出现停工时间 (Idle time), 这些停工时间会在一定程度上给系统流程时间、库存和订单延迟带来负面影响。Park 和 Bobrowski^[52] 阐明了 Job shop 下的工作分配机制, 在同时考虑工作特征和生产单元属性的情况下确定各工作的投放时间, 论文指出了劳动力柔性对工作投放过程的重要作用。Cigolini 等人^[53] 提出了一种新的方法模式, 通过对比动态不确定环境下的多个订单检查与投放 (ORR) 模型来评定订单检查与投放 (ORR) 技术的有效性。论文通过对先前文献中开发的三个订单检查与投放 (ORR) 模型进行充分比较, 研究了三方面问题: (1) 动态不确定环境对性能的影响; (2) 面对 Job shop 实际生产环境中的需求波动时三种模型的健壮性; (3) 工作量造成生产延期完成方式的整体有效性。

在单元化生产系统中, 生产任务分配和生产调度是生产控制决策中考虑的两个重要问题^[54]。单元化生产下的生产任务分配问题研究如何将产品订单分配给建立好的生产单元, 分配方案中包括每个生产单元加工的产品种类和数量, 甚至具体到每个机器上加工的产品种类和数量。单元化生产下的生产任务分配问题包括三个主要决策, 即选择产品、选择生产单元和确定该产品在生产单元生产任务列表中的次序和任务量。产品导向型分配策略通常先选择产品, 然后在该产品的可用生产单元中选择一个生产单元, 而生产单元导向型策略通常先确定生产单元, 然后根据生产单元的加工能力, 确定分配哪些产品。生产任务分配法则的好坏从多个方面直接影响生产系统的性能, 比如生产成本、延迟时间/费用、柔性等。单元化生产下的生产任务分配问题通常使用的目标函数包括: 在制品库存最小化, 生产延期时间最小化, 生产单元能力利用率最大化, 和生产单元间工作量均衡水平最大化。

与六种生产调度启发式算法相结合, Mahmoodi 等人^[55]对比了单元化生产系统下三种订单投放启发式算法和两种到期日分配启发式算法。Süer 和 Ortega^[56]以最大完工时间和生产单元内人员移动次数最小化为目标, 建立了一个解决生产任务分配问题的启发式算法。Davis 和 Mabert^[57]建立了两个数学规划模型来分别解决独立生产单元和内联式生产单元两种不同的单元化生产系统背景下的订单分配和工人分配/再分配决策。为解决劳动密集型生产单元下的生产任务分配和产品排序问题, Süer 等人^[58]提出了一个包括工人分配、生产任务分配和产品排序三个阶段的方法。作为前期研究工作的扩展, 他们提出了两个解决生产任务分配问题的数学模型, 模型以工人移动最小化为目标函数, 并将产品排序问题视为旅行商问题 (TSP)。

以上研究大都将生产任务分配问题与工人分配问题同时考虑, 也有一些研究专门致力于解决生产任务分配问题。文献【59】比较了多个生产任务分配法则和方法, 并对它们各自的优缺点进行了总结归纳。文献【60】、【61】通过仿真实验研究了影响单元化生产系统生产任务分配问题的因素。Espino^[62]从产品优先和生产单元优先两种不同的角度提出了四种生产任务分配算法。Süer 等人^[63, 64]详细讨论了单元化生产系统下的生产任务分配法则, 同时分别针对内联式生产单元和独立生产单元提出了不考虑工人分配情况的生产任务分配算法。Süer 等人^[65]从最终完工时间的角度比较了三种生产任务分配启发式算法, 一个生产单元调度启发式算法和两级数学模型。

2.3.2 决策要素

(1) 批量分割

批量分割是指将加工周期过长的批次拆分为若干较小的批次, 使该批产品可以同时多个生产单元中加工, 它能够有效解决生产单元可用生产能力小于产品所需能力的问题。在批量不可分割的问题中, 通常要利用各种约束和分配规则来保证一批产品能够完整地分配到一个生产单元中且该生产单元也具备生产该批产品的能力。而在批量可分割的问题中, 分配规则就相对简单, 只需按顺序来分配产品, 当一个生产单元的生产能力不足时, 就将一批产品分割为两批, 并将分割出的产品分配给其他可用生产单元。在一些分配法则下, 通过批量分割将加工周期较长的任务拆分为若干加工周期较短的任务后再将其分配到多个生产单元中能够有效减少最大完工时间, 这样做虽然增加了换装时间, 却能保证产品在规定的交付期之前完成生产, 从而减少延迟。

Süer 等人^[65]提出了三种不同的启发式算法来解决面向劳动密集型生产单元的生产任务分配问题, 其中两种算法都允许对生产任务过大的产品族进行拆分。两种算法中采用了不同的分割原则, 一种是在所分配生产单元的生产能力余额不足时, 对子产品族进行拆分; 另一种是根据总工作量和生产单元数计算出各生产单元的最佳完工时间, 优先分配加工周期小于最佳完工时间的产品族, 将加工周期大于最佳完工时间的产品族进行拆分后, 再将其分配到合适的生产单元, 分配时要保证各生产单元内生产任务的累积加工周期小于最佳

完工时间。当生产任务分配问题的目标函数是减小最大完工时间时,最好采用第二种批量分割原则。此时,完成所有生产任务所需的最大完工时间与给定的最佳完工时间成正相关关系。Süer 等人^[56]在解决生产任务分配问题时分别考虑了批量可分割和不可分割两种情况,同时他们还指出在完成同样的生产任务时,批量不可分割情况下的最大完工时间要大于批量可分割情况下的。

(2) 人员移动次数

当生产单元中的设备较为简易且差异不大时,生产单元中的工人,尤其是多能工,对生产单元的生产能力起决定性作用。对于这种生产单元,当有新产品分配到生产单元后,通常不会改变原生产单元的构成,而是对生产单元内的工人进行调整。Süer 和 Dagli^[66]在解决劳动密集型生产单元的生产任务分配问题时,利用人员移动矩阵表示每两种产品间的人员移动距离,并借助旅行商问题的解决思路找出总移动距离最小的产品排序方案。Süer 等人^[56]使用一个三阶段方法同时解决了劳动密集型生产单元下的生产任务分配和产品排序问题,该问题的目标函数是最大完工时间、机器总数和生产单元内人员移动次数最小化。该方法首先为每个产品的工序分配最佳人选并评价产品间的相似程度,接着以减少最大完工时间和机器总数为目标分配生产任务,最后,以减少生产单元内人员移动次数为目标对各生产单元内分配的产品进行排序。

(3) 交付期 (Due date)

交付期根据客户订单规定的交货日期确定,在分配生产任务和安排生产顺序时要保证所有产品都能在交付期之前完成生产。如果安排的生产任务无法在交付期之前完成生产,就会导致延迟交货,同时给企业带来经济和声誉上的损失。对于考虑交付期的问题,必须同时考虑产品的换装时间、加工时间等,只有准确计算产品的加工周期,安排合理的生产次序,才能保证产品按时完成生产。为保证所有产品在交付期之前完成生产,通常将待分配的生产任务按照交付期最早优先 (Earliest Due Date, EDD) 原则排序,则交付期最早的产品将被优先安排分配。Süer 等人^[67]在解决以减少延迟工作数量为目标的生产任务分配和排序问题时,考虑了交付期对分配方案的约束。

2.3.3 决策方法

面向生产单元的生产任务分配问题通常有两种解决方法,一种是整体方法,这种方法将生产任务分配问题构建成为一个大的混合整数线性规划 (MILP) 问题,然后再用启发式程序来解决;另一种方法是层次化方法,这种方法将整个生产任务分配问题划分为若干具有层次关系的小问题^[68]。

在第一种方法中,常用的启发式程序包括启发式算法和元启发式算法,这两类算法在处理复杂问题时能够在较短的时间内得到较好的结果。考虑到生产排序问题通常需要较快的响应速度, Babayigit^[69]使用遗传算法和数学模型同时解决了工人分配和生产任务分配问题,通过对多个算例求解证实在处理规模较大的问题时遗传算法能够很快地得出最优或

近似最优解。Süer 等人^[67]结合遗传算法和局部最优化方法提出了三种能够同时解决生产任务分配和排序问题的快速有效的方法。其中,遗传算法被用来生成工作序列,局部最优化方法用于分配生产任务并减少延迟工作的数量。Yarimoglu^[70]利用改进的 p-中值模型来解决在人员限制下面向同期化生产单元的生产任务分配问题,该问题在分配生产任务时以提高各生产单元内产品的相似度为目标,但通过实验证实,这样并不利于产品排序。

最早的层次生产计划(HPP)是由 Hax 和 Meal^[71]提出的,他们的方法将所有产品划分为三个等级,第一级对于不同的类使用由整体计划模型衍生出的线性规划模型,每类包含类似季节需求模式和生产率的内容,第二级使用启发式解集将这些类转换为族,族里包含类似换装成本的项,第三级基于均衡到期时间将族分解为若干项。这种方法从最高级别开始按顺序进行决策,每一级的决定都会成为下一级的约束条件。大多数层次生产计划方法都假设生产资源具备无限大的生产能力,从而忽略了生产能力约束,然而,对于存在瓶颈工序的问题来说,这样会降低解的可行性。Akturk 和 Wilson^[68]提出了一种层次生产任务分配方法来解决单元化生产系统的生产计划制定问题,该方法以实现可变生产成本最小化为目标,同时考虑了生产与库存间平衡、可用生产能力等约束条件。

生产任务分配和排序问题的一个重要特征就是在处理实际问题时有很多参数都是难以准确计算的,如加工时间、产品需求、可用生产能力等,因此,通常要借助模糊集合论来处理这些无法精确给出的数据。自 Bellman 和 Zadeh^[72]介绍了模糊决策和 Zimmerman^[73]建立了模糊数学规划问题后,出现了大量模糊最优化领域的方法。模糊数学规划是解决多目标最优化问题的有效方法,尤其是在目标函数和约束条件的模糊愿望水平存在时。在模糊数学规划中,模糊决策是由用来定义目标和约束条件间关系的模糊算子确定的。早在二十世纪七十年代末就已经有研究将模糊集合理论应用于生产排序问题。模糊概念被用来表示某些不确定的时间参数,如模糊交付期、模糊加工时间等。Süer 等人^[74]利用四个不同的双目标规划数学模型解决了不同生产环境下的劳动密集型生产单元生产任务分配问题。他们结合模糊集合论的知识利用两种不同的模糊算子将其中两个数学模型转换为模糊数学规划问题,还指出由于每个产品都有多个可以分配的生产单元,生产任务分配问题与并行机调度问题相似。Süer 等人^[75]研究了劳动密集型生产单元下的模糊双目标生产任务分配问题,并检验了应用不同模糊算子对模型的影响。该问题能够同时确定生产单元数量、生产单元大小、产品分配方案和产品生产顺序。

2.3.4 评价指标

(1) 最大完工时间(Makespan)

最大完工时间是指整个生产系统在完成一系列工作时从工作开始到全部工作完成期间间隔的时间。在单元化生产系统中,最大完工时间就是指在给定计划期内完成指定工作时,从最早投入生产的生产单元的开始时间到最晚结束的生产单元的结束时间中间所花费的时间。过多的产品批次或产品族会带来大量换装时间,因此对减小最大完工时间具有负

面影响。同时,由于不同的加工顺序会产生不同的换装时间,只有在对各生产单元内的生产任务排序后才能得到生产所有产品所需的准确的完工时间。在处理生产任务分配问题时,可以通过数学建模和启发式算法减少最大完工时间,常用的启发式算法包括加工时间最长产品优先分配算法、产品优先分配给生产任务最少的生产单元、McNaughton 算法。

(2) 延迟

在实际生产过程中,无论是提前完成生产或延迟完成生产都会给企业带来损失,但相比提前完成生产带来的库存费用,延迟带来的不良影响更大。延迟是以交付期为导向的系统性能评价指标,当产品无法在交付期之前完成生产时就会发生延迟,生产任务分配和排序问题中通常使用延迟工作的数量、总延迟时间、和最大延迟时间来衡量延迟程度。虽然在生产任务分配和排序问题中以减少生产任务延迟为目标的问题比较少,但由于生产任务延迟对客户满意度具有很大影响,因此常被作为衡量生产任务分配和排序工作的重要指标。当工作延迟时,会伴随着出现多种费用,如固定惩罚成本、当部分产品未能按时产出时产生的可变惩罚成本、销售损失等。通过合理安排生产任务次序和采用允许批量分割的分配法则,能够有效减少延迟完成的工作数量。例如,按照交付期最早产品优先(EDD)原则对产品进行排序就能有效减少延迟任务数量和延迟时间。延迟任务数量与生产单元的数量和生产单元中人员数量成负相关关系,但增加人员数量会带来用工成本的大幅增加。

2.4 本论文所用方法概述

2.4.1 非支配排序遗传算法

遗传算法(Genetic Algorithm,GA)是一种能够有效解决组合优化问题的元启发式算法,由 John Holland^[76]于1975年提出。遗传算法通过在随机产生的具有一定结构的解集上应用适者生存法则来模拟自然界中生物的进化、信息交换和创造子代过程来寻找问题的最优解。遗传算法被广泛地应用于很多领域,包括工程、生物、经济、农业、商业、电子通信和制造业^[77]。在现有文献中,标准遗传算法以及各种改进后的遗传算法已经被有效地应用于生产单元构建问题。其中,由 Falkenauer^[78]提出的分组遗传算法(Grouping Genetic Algorithm,GGA)是一种更适用于分组问题的算法,该算法对经典遗传算法的编码机制和遗传算子进行了适当修改。

遗传算法通过使用模拟进化过程(如适者生存战略)来搜索方程的解空间。通常情况下,每一代中适应值最高的个体可以进行繁殖并存活到下一代,以此不断改善后续各代的质量。但是,适应度较差的个体也不会全部被抛弃,它们中的一部分也会偶然被选作父代进行繁殖并存活到下一代。研究证实,在解决线性和非线性问题时,遗传算法能够探索到解空间的所有区域并能够通过种群中的个体使用变异、杂交和选择算子来深度开发那些可能存在最优解的区域^[79]。遗传算法最大的优势是其应用不受目标函数和约束条件的限制。标准遗传算法的流程如图2-4所示。

1. 给代数计数器赋值, $i=0$ 。
2. 随机产生 N 个个体, 创建初始种群 $P(i)$ 。
3. 根据种群中每个个体所代表的解求得解对应的目标函数值, 再根据目标函数值确定每个个体的适应值。
4. 进化过程向前推进一代, $i=i+1$ 。
5. 依据适应值从上一代种群 $P(i-1)$ 中随机挑选出 N 个个体, 建立新的种群 $P(i)$ 。
 - (a) 从新的种群中随机选择 R 个父代, 并通过遗传算子产生新的子代。
 - (b) 根据目标函数确定评估新产生的子代的适应值。
6. 如果 i 小于设定的最大遗传代数, 执行第 4 步。
7. 给出算法得到的最优解。

图 2-4 标准遗传算法流程

Fig. 2-4 The procedure of classical genetic algorithm

遗传算法属于元启发式算法中的进化算法, 由于它能够解决生产单元构建这种具备 NP 困难属性的组合优化问题, 并在合理的时间内得到全局或近似全局最优解, 因此它在过去的研究中得到了广泛的应用, 并取得了令人满意的结果。与其他元启发式算法相比, 遗传算法的搜索过程基于一个种群的解, 而非单一解; 它从解空间的不同地方开始搜索最优解, 能有效避免陷入局部最优解的情况发生; 最终得到多个解, 决策者可以灵活选择最适合的解^[80,81]。Onwubolu 和 Mutingi^[82] 在考虑实施单元化生产过程中遇到的实际问题的情况下, 利用遗传算法解决了生产单元构建问题。James 等人^[83] 提出了一个解决生产单元构建问题的混合分组遗传算法, 该算法将局部搜索方法与标准分组遗传算法有效结合。Pitombeira Neto 和 Gonçalves Filho^[84] 通过一个多目标优化方法来解决生产单元构建问题, 并用遗传算法求得该方法的一组帕累托最优解。

对于多目标优化问题, 也存在大量有效的遗传算法, 比如非支配排序遗传算法-II (NSGA-II)、强度帕累托进化算法 (SPEA)、帕累托存档进化策略 (PAES)、多目标微分进化算法 (MODE)、向量评价遗传算法 (VEGA)、小生境帕累托遗传算法 (NPGA) 等。其中 Deb 等人^[85] 提出的非支配排序遗传算法-II 较为常用, 该算法在非支配排序遗传算法 (NSGA) 的基础上进行改善, 克服了原有算法非支配排序过程计算复杂度较高、缺乏精英意识和需要确定小生境半径的缺陷。非支配排序遗传算法能够同时处理多个目标函数, 并给出一组同时满足多个目标函数的非支配解。

非支配排序遗传算法-II 的标准算法流程如下^[85]: 首先, 随机产生一个个体数为 N 的初始种群 P_0 ; 根据适应函数计算初始种群中每个个体的适应值, 对所有个体进行两两对比, 根据适应值和拥挤距离判断它们间的支配关系并将所有个体划分为不同的边界; 根据非支配等级, 给每一个边界里包含的个体分配一个级别, 该级别与其非支配等级相一致; 级别为 1 的个体属于最高等级, 它们不受其他任何个体的支配, 级别为 2 的个体仅被级别为 1 的一些个体支配, 以此类推; 然后, 对排序后的个体进行选择、杂交和变异, 得到个

体数为 N 的子代 Q_0 ; 将父代 P_0 和子代融合为一个个体数位 $2N$ 的整体 C_0 ; 对 C_0 再次进行非支配排序, 将排序后的前 N 个个体取出, 得到新一代个体数为 N 的父代 P_1 ; 重复以上过程, 直至满足停止条件时结束进化过程。

2.4.2 启发式算法

启发式是一种用于解决、学习和发现问题的基于经验的技术。在无法使用枚举搜索方法的时候, 可以利用启发式方法加快寻找满意解决方案的过程。启发式方法的思路通常来自于经验法则, 依据经验、专业知识等猜测, 直观判断, 或常识。更准确地说, 启发式方法是那些使用易获取的信息来控制问题解决过程的战略, 尽管那些信息的适用性较低^[16]。

启发式方法是简单有效的法则, 是演化过程的硬编码, 它解释了人们是如何制定决策、做判断和解决问题的, 尤其是面对那些复杂的、信息不完全的问题时。这些法则在大多数情况下都表现良好, 但在一些情况下则会导致系统误差或认知偏差。尽管 Amos Tversky 和 Daniel Kahneman 在从人类决策者身上探索启发式方法上做了大量工作, 但启发式这一概念最早是由诺贝尔奖得主 Herbert Simon 提出的。

在计算机科学领域, 启发式方法是一种在解决问题时忽略所得到的解是否能被证实是正确解的技术, 但是它通常能够得到较好的解, 也可以解决那些包含较复杂问题的解的简单问题。在牺牲解的准确性和精确度的前提下, 启发式方法能够获得较好的计算表现或概念上的简化。启发式方法可以通过使用查找树来完成任务, 但它并不会产生所有可能的解分支, 而是选择那些更有可能产生结果的分支。

启发式算法之所以经常被使用, 是因为它可以在不需要用数学方法去证明满足一组给定要求的情况下工作。若使用启发式方法去满足给定要求, 那么当工程师或设计者使用的数据组对表达未来系统状态没有帮助时, 启发式方法存在的隐患就会暴露出来。启发式算法不是每次都能在以较快的速度得到较好的解, 且不同的问题所采用的算法也不同, 故这种方法的通用性较差。

启发式算法是相对于最优化算法提出的, 它是基于经验或直观判断构建的算法。启发式算法能够在所有可能的解中找到问题的一个可行解, 但是并不能够保证所找到的解是最优解, 且该可行解与最优解直接的差距是不确定的, 因此, 它通常被认为是一种近似的、不准确的算法。这种算法通常能够用简单的方法在较短的时间内找个一个非常接近最优解的近似最优解。启发式算法有时也是准确的, 也就说它确实找到了问题的最优解, 但是在这个最优解能够被证实确实是所解问题的最优解之前, 这些算法仍要被称作是启发式的。启发式算法中使用的方法都是已知的方法, 如贪婪算法和局部搜索算法, 但是为了追求简化和速度, 启发式算法会忽略甚至隐藏问题的一些需求。

3 面向 Seru 生产的生产单元构建决策研究

3.1 生产单元构建问题的研究进展与本论文的注目点

无论是对传统单元化生产还是对 Seru 生产, 国际上的学术研究大致都从四个主要方面展开, 即生产单元构建问题, 布局设计问题, 生产计划制定问题, 和生产调度与控制问题^[67]。生产单元构建问题的解决是研究其他问题的基础, 对整个单元化生产系统的性能具有决定性影响, 因此, 这一直是学术界关注的热点问题。对于拥有较长历史, 研究内容体系相对完善的传统单元化生产来说, 生产单元构建问题的研究已取得了丰硕的成果。

传统单元化生产下的生产单元构建问题大都基于机器-产品(零部件)关联矩阵, 这种矩阵反映了建立生产单元和产品(零部件)族所需的重要信息, 即产品的加工需求和机器的加工能力。早期, 机器-产品(零部件)关联矩阵中元素的值只有 0 和 1 两种情况, 若矩阵 A 中的机器 i 能够加工产品 j, 则 $a_{ij}=1$, 其他情况下, $a_{ij}=0$ 。对于这种矩阵, 利用聚类算法调整矩阵中行和列的位置, 使矩阵转化为分块对角矩阵并确保将所有元素都划分到分块中, 那么, 每一个分块所对应的机器组成一个生产单元, 相应的产品(零部件)组成一个产品(零部件)族, 无法放入分块内的元素就称为例外元素。这种生产单元构建问题考虑因素较少, 求解过程相对简单, 通常以整个生产系统内的相似性系数值最大或生产单元间移动次数最少(例外元素数量最小)为目标, 但由于这种矩阵中没有考虑换装时间、加工时间、加工顺序和生产量, 常会出现机器间和生产单元间工作量不均衡。生产单元构建问题中常用的聚类算法包括键能算法(Bond energy algorithm)、排序聚类算法(Rank order clustering)、排序聚类算法 2(Rank order clustering 2)、改进排序聚类算法(Modified rank order clustering)、直接聚类算法(Direct clustering algorithm)等。

随着聚类算法在生产单元构建问题上的应用逐渐成熟, 为了提高所研究问题的实用性, 越来越多实际生产要素被融合进生产单元构建问题中。与早期研究的生产单元构建问题相比, 考虑实际生产要素的生产单元构建问题在数据、目标函数和约束条件上都有很大改进。生产单元构建问题中最常考虑的生产要素包括加工时间、加工顺序、可替换工艺路线、同质机器、产品需求量、生产单元大小、机器加工能力等。对于考虑可替换工艺路线、加工顺序的生产单元构建问题, 机器-产品(零部件)关联矩阵中每个产品(零部件)可以对应多列, 每一列表示产品(零部件)的一种工艺路线, 矩阵中元素的值可以是任意实数, 元素值的大小表示某一机器在某一产品(零部件)的一种工艺路线下加工的工序编号。除了机器-产品(零部件)关联矩阵外, 考虑多个生产要素的问题可能需要的数据包括每种产品(零部件)的需求量、每台机器加工指定工序的时间、机器的最大生产能力、不同类别机器的数量等。这一阶段使用最多的目标函数是生产单元间移动次数(费用)最小化, 也有一些研究选择工作量均衡作为评价指标。由于考虑因素较多, 问题中也相应地增加了对机器数量、机器能力、生产单元大小、生产单元数量等条件的约束。考虑多个生产要素的问题复杂性大大提高, 仅通过对矩阵进行调整已经无法得到解, 因此, 这类问题的解决

方法大都是先根据问题建立数学模型,然后根据数学模型的特征,选择合适的软件或算法来对数学模型求解,得到生产单元构建方案。

上述研究解决的都是静态环境下的生产单元构建问题,通常忽略了不同时间段需求变动所带来的产品品种和需求量的变化。静态环境下市场需求是确定的、已知的、长时间内不会发生变化的,因此,这种问题下得到的生产单元构建方案只能适应问题中给定的需求信息且在长时间内不会发生变动。若需求发生变化,原来的方案就有可能不能满足生产需求。近几年,已经有一些研究者开始关注动态环境下的生产单元构建问题,并提出了各种应对方法。Vakharia 和 Kaku^[88]在处理产品需求变化时,保持生产单元的组成和布局不变,只改变产品族的结构,这样就避免了生产单元重组带来的费用。Harlahakis 等人^[89]从增强生产单元构建方案的鲁棒性入手解决动态需求环境下的生产单元构建问题,他们主张在设计生产单元时尽量满足多个计划期的需求,这样也可避免生产单元的频繁重组。

更多研究者从改变生产单元组成和形态入手研究动态需求环境下的生产单元构建问题。对于需求可以预测的市场环境,一些研究者提出了不同计划期内采用不同分组方案的动态生产单元构建问题。动态生产单元构建问题通常包括多个计划期,每个计划期内的需求都会发生变化,但是可以通过预测得到,然后根据需求的变化情况给出若干种生产单元构建配置方案,最后在考虑加工需求、各配置方案的生产能力、每一计划期内的加工成本以及各配置方案间转换成本的基础上,确定每个计划期内应该选择的生产单元构建配置方案。若在不同生产单元分组方案间转换时不发生成本或成本可以忽略,则各个计划期内采用当期内的最佳方案就可保证多个计划期内整体最优。实际上,由于从前一阶段的最佳配置方案向下一阶段的最佳配置方案转换时可能产生由设备的拆除、搬运、和安装带来的转换费用、损失生产时间带来的费用和再学习费用,而在两种不同的配置方案间转换又会产生不同的转换费用。在重组生产单元之前必须要进行成本收益分析,如果使用了错误的配置方案将会导致以后各期的成本大幅增加。因此,动态生产单元构建问题的难点在于各个计划期内选择什么样的配置方案才能使整个大的计划期内得到的配置方案整体最优,也许最终方案中各个计划期内使用的是次优配置方案。

对于需求不可预测、完全不确定的市场环境,一些研究者提出了虚拟生产单元的概念。虚拟生产单元是将生产系统中的资源从逻辑上进行分组,这些资源不必在车间里来回移动组合成生产单元形态,而是暂时用来共同加工产品(零件),这样虽避免了生产单元重组带来的巨额费用,却会造成物料移动次数和距离的增加。生产任务完成后,虚拟生产单元内资源间的关系将被打破,需求发生变化后,这些资源又根据新的加工需求重新联系起来。在机器体积较大,不易移动的情况下,使用虚拟生产单元既能保证生产柔性,又能避免重组带来的麻烦。与传统形态的生产单元相比,采用虚拟生产单元能够更好地应对需求波动,获得更高的柔性。动态生产单元构建问题和虚拟生产单元构建问题是生产单元构建这一研究领域里的前沿问题,目前只是提出了相关的思想和理论,具体解决方法尚在研究之中。

可以看出,传统单元化生产方式下生产单元构建问题的研究已经比较系统,研究成果

也很丰硕。对于刚刚兴起的 Seru 生产来说,大部分研究成果都停留在对企业实施成功经验介绍的层次,还并没有深入到解决 Seru 生产实施的具体决策。同时,由于 Seru 生产与传统单元化生产之间存在本质区别,故将用于解决传统单元化生产的生产单元构建问题的解决方法直接套用在 Seru 生产上是不现实的,也是不科学的。因此,亟须深入研究面向 Seru 生产的生产单元构建决策,在借鉴传统单元化生产的生产单元构建决策解决思路的基础上,结合 Seru 生产的特征找到适宜的理论与方法来解决该问题。

本论文重点研究面向 Seru 生产的生产单元构建决策,该决策必须满足 Seru 生产下生产单元完结性(Kankatsu)这一显著特征。完结性是指一个产品(零部件)的整个生产过程都在一个生产单元内完成,也就是说,生产过程中不会发生生产单元间物料移动,各生产单元是相互独立的^[90]。复制机器、外包产品(零部件)和寻找可替换的工艺路线/机器是三种减少生产单元间物料移动的有效方法^[91]。虽然三种方法都能有效降低生产单元间物料移动成本,但它们分别会导致设备购买费用、外包费用和更高的加工成本。大多数生产单元构建问题都假设每类机器的数量为 1,由于 Seru 生产较多应用于电子产品制造行业,所生产的产品大多是体型小、重量轻、易于装配的产品,同时由于注重多能工的手工加工能力,其使用的设备大部分也是简单的、价格低的、易购买(复制)的,因此,本文考虑了同质机器因素,通过将这些复制得到的同质机器分配到所有需要它们的生产单元内,可以彻底消除瓶颈,从而消除生产单元间物料移动。为了更好地实现生产单元的完结性,本论文研究的问题还考虑了可替换工艺路线因素,产品可以根据机器能力和加工时间在多条工艺路线中选择最合适的一条。

本章研究面向 Seru 生产的单元构建问题,在考虑同质机器和可替换工艺路线的情况下,建立了一个以实现生产单元间工作量均衡为目标的非线性规划数学模型。与以前的研究不同,本论文不以生产单元间物料移动和相似性系数为系统性能衡量标准,而是利用时间参数来评价生产单元间的工作量均衡程度。问题中考虑了多个实际生产要素,包括加工时间、换装时间、加工顺序、产品需求量、机器加工能力、生产单元大小和生产单元数量。考虑到所建数学模型是多目标和非线性的,本章提出了一个基于非支配排序遗传算法-II 的混合遗传算法对数学模型进行求解,并用算例证实了数学模型的正确性和算法的有效性。

3.2 面向 Seru 生产的生产单元构建问题描述

本论文研究的面向 Seru 生产的生产单元构建问题主要解决在静态需求环境下,当考虑同质机器和可替换工艺路线时如何构建相互独立的、生产单元间工作量均衡的完结型生产单元。同质机器和可替换工艺路线是保证生产单元完结性和独立性的重要因素,它们可以彻底消除生产单元间物料移动,确保每个生产单元内生产的产品的所有操作都可以从头至尾在该生产单元内完成。本问题的已知信息包括不同类型产品的需求量、各类机器包含的同质机器总数、各类产品的工艺路线及加工时间、以及不同产品类型间的换装时间。需

要构建的生产单元数量没有作为已知条件给出,而是作为决策变量根据生产系统中机器的数量和实际加工需求确定的,被分配到一个生产单元中的不同批次的产品组成一个产品族,图 3-1 形象地描述了本论文所研究的生产单元构建问题。

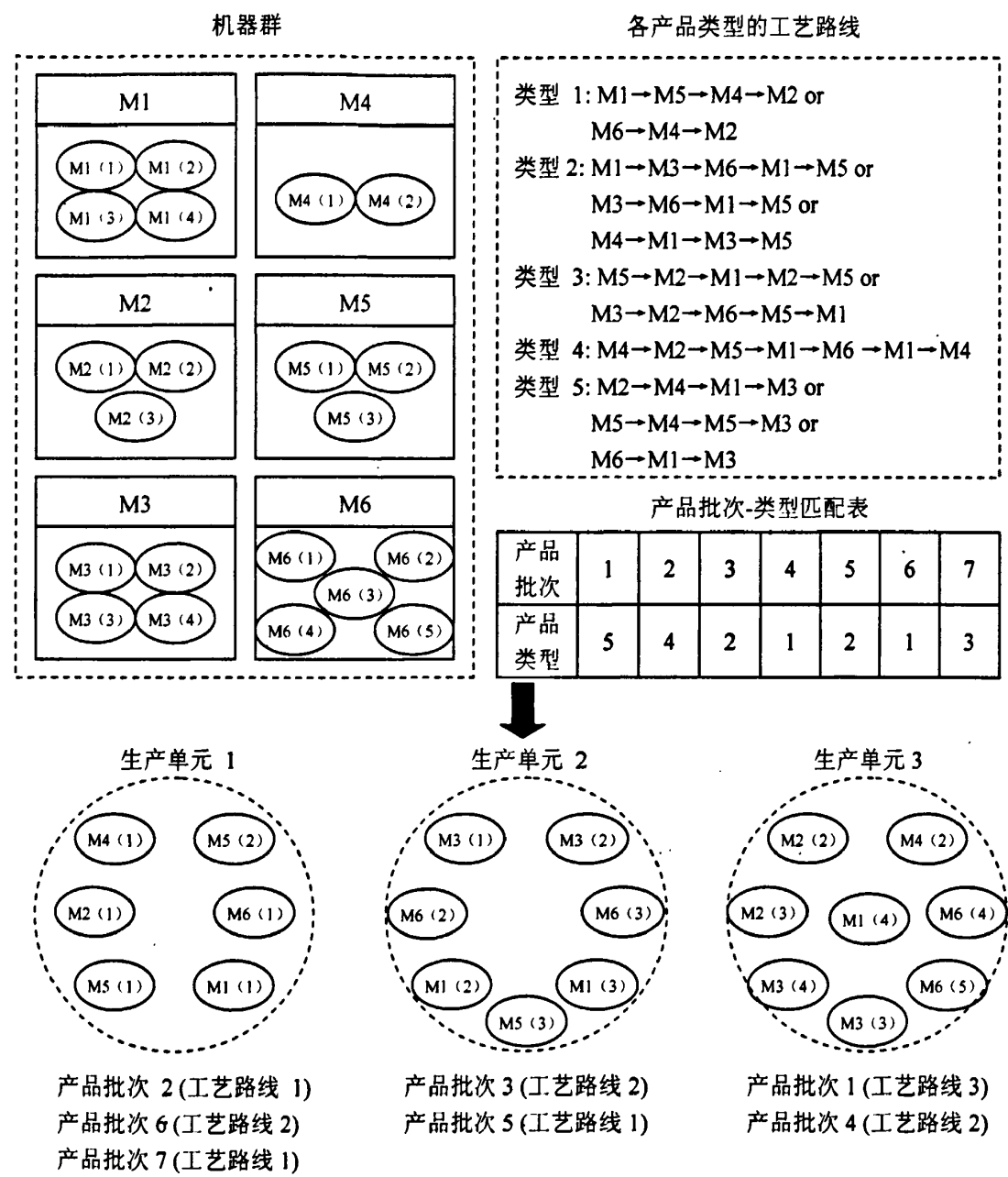


图 3-1 生产单元构建问题示意图

Fig. 3-1 The schematic diagram of the seru formation problem

从图中可以看出,每类机器都包含多个同质机器,这些同质机器的加工能力完全一致,每类产品拥有至少一条工艺路线,不同的生产批次可以包含相同的产品类型,通过对这些资源进行合理配置,最终形成三个生产单元。本问题中包含的几项重要决策包括:

- (1) 整个 Seru 生产系统中需要建立的生产单元的数量;
- (2) 分配给各个生产单元的机器类型;

- (3) 各个生产单元内分配的每种机器类型的同质机器数量;
- (4) 分配给各个生产单元的产品批次;
- (5) 各批次产品选择的工艺路线;
- (6) 每台机器上分配的工作量。

本论文研究的生产单元构建问题以生产单元间的工作量均衡程度来衡量系统性能,因为对于完结型生产单元来说,不会发生生产单元间物料移动,而工作量均衡程度却对整个生产系统的时间框架和生产同期化水平具有很大影响。如果工作量分配不均衡,就有可能造成较长的机器停工时间、较低的生产单元能力利用率和部分生产单元过载现象。可替换工艺路线和同质机器能够明显缓解工作量分配的不均衡程度,改变产品的加工路线可以将过载的加工任务分配给未被充分利用的机器,而增加同质机器的投入量可以帮过载机器分担工作量,最终实现整个生产系统中各生产单元的平衡。工作量分配均衡程度较高的生产系统通常具备较短的最大完工时间 (makespan), 最大完工时间的长短会直接影响产品的交货时间,尤其是在多个具有相同交货时间的产品批次分配到不同生产单元生产时,生产单元间工作量均衡可以缩短各个生产单元完工的时间间隔,这样不但大大节省了提早下线产品的库存成本,还能够保证产品订单的准时交货。图 3-2 描述了工作量均衡对最大完工时间的影响。

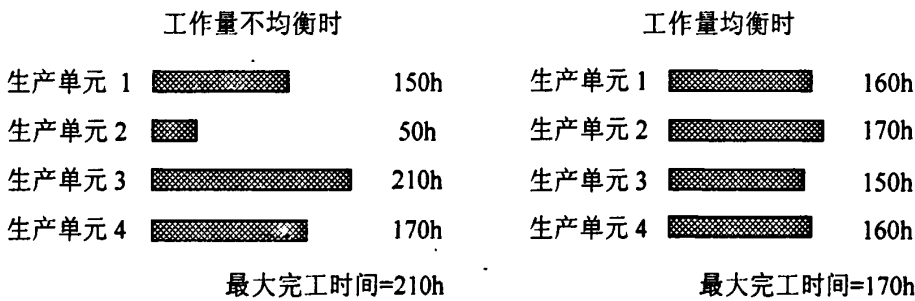


图 3-2 工作量均衡对最大完工时间的影响
Fig. 3-2 The influence of workload balance on makespan

3.3 面向 Seru 生产的生产单元构建决策模型

本节建立了一个非线性数学规划模型来解决具备完结性的生产单元构建问题,数学模型的目标函数分别是所有生产单元生产总流程时间的均值和各生产单元生产总流程时间的方差最小化,其中,流程时间是指加工周期和换装时间的总和。流程时间代表分配给生产单元的工作量,当每个生产单元的总流程时间趋向一致时就意味着分配给各生产单元的工作量趋于均衡。为了增加问题的实用性和适用性,模型中将生产单元大小、可用同质机器数量、可替换工艺路线、机器加工能力和产品需求量等要素作为主要约束条件。

3.3.1 问题假设

- (1) 市场需求是确定型的,根据交付期和需求量,各类产品的需求事先被划分为若干生产批次,每个生产批次中只能包含一类产品。

- (2) 产品批量不可分割, 每个产品批次的生产任务必须在同一个生产单元内完成。
- (3) 每类机器只能从事一种类型的作业, 各类机器完成给定工序的加工时间都是已知的、确定的。
- (4) 在一个产品的一条工艺路线下, 每类机器只能负责至多一道工序。
- (5) 属于同一种类型的同质机器能够以相同的加工时间完成相同的工序。
- (6) 一个生产单元内可以分配多个属于同一种类型的同质机器。
- (7) 生产单元的最大机器容纳能力要确保生产单元能够容纳生产任意类型产品所需的机器种类数。

3.3.2 符号定义

- 输入参数:

- c 生产单元编号, $c=1,2,\dots,C$;
- p 产品类型编号, $p=1,2,\dots,P$;
- m 机器类型编号, $m=1,2,\dots,M$;
- N_m m 类机器包含的同质机器数量;
- $m(i)$ m 类机器下同质机器编号, $m(i)=m(1),m(2),\dots,m(N_m)$;
- b 产品批次编号, $b=1,2,\dots,B$;
- r 工艺路线编号, 当产品类型 p 时, $r=1,2,\dots,R_p$;
- s 工序编号, 对于 p 类产品的工艺路线 r , $s=1,2,\dots,S_{p,r}$;
- $e_{(p,r)}$ p 类产品的工艺路线 r 中各工序对应的机器类型集合;
- $A_{e_{(p,r)}}$ 集合 $e_{(p,r)}$ 的势;
- V_b 产品批次 b 的批量;
- $t_{s,m}^{p,r}$ p 类产品的工艺路线 r 中第 s 道工序在 m 类机器上的加工时间;
- $t_{b',b}$ 从产品批次 b' 到下一个产品批次 b 的换装时间;
- N 每个生产单元中能够容纳的机器数量上限;
- G 在给定的时期内每台机器的可用生产能力上限。

- 辅助二进制变量:

$$U_{m(i)}^c = \begin{cases} 1, & \text{当编号为 } m(i) \text{ 的机器分配到生产单元 } c \text{ 时;} \\ 0, & \text{其他。} \end{cases}$$

$$X_b^c = \begin{cases} 1, & \text{当产品批次 } b \text{ 分配到生产单元 } c \text{ 时;} \\ 0, & \text{其他。} \end{cases}$$

$$Y_b^p = \begin{cases} 1, & \text{当产品批次 } b \text{ 中的产品属于产品类型 } p \text{ 时;} \\ 0, & \text{其他。} \end{cases}$$

$$H_b^r = \begin{cases} 1, & \text{当产品批次 } b \text{ 按照工艺路线 } r \text{ 组织生产时;} \\ 0, & \text{其他。} \end{cases}$$

$$D_{s,m}^{p,r} = \begin{cases} 1, & \text{当 } p \text{ 类产品的工艺路线 } r \text{ 中第 } s \text{ 道工序在 } m \text{ 类机器上加工时;} \\ 0, & \text{其他。} \end{cases}$$

$$W_c^{b',b} = \begin{cases} 1, & \text{当分配到生产单元 } c \text{ 中的产品批次 } b \text{ 和它的前一个产品批次 } b' \text{ 属于同一个产} \\ & \text{品类型时;} \\ 0, & \text{其他。} \end{cases}$$

• 决策变量:

C 需要建立的生产单元数;

$V_{m(i)}^b$ 产品批次 b 分配到编号为 $m(i)$ 的机器上的产量;

F_b^c 当产品批次 b 在生产单元 c 中生产时的加工周期;

• 输出参数:

Γ 生产单元构建方案。

3.3.3 数学模型

本论文研究的生产单元构建问题以生产单元间工作量均衡为建立生产单元和衡量生产系统性能的标准。数学模型中,工作量由生产单元完成生产任务所需的总流程时间表示,工作量均衡程度依据各生产单元生产总流程时间的均值和方差来衡量。为了能够确定生产单元的最佳数量与机器和产品批次的合理分配方案,并实现生产单元间工作量均衡,本节建立了一个非线性规划数学模型。

由于一个生产单元中可以分配多个属于同一机器的同质机器,这些机器具有相同的加工能力,它们可以在相同的加工时间内并行加工同样的工序,这样能够大大缩减工序的单位加工时间,提高产品的单位时间产出量。对于产品批次 b 来说,如果它属于产品类型 p ,且被分配到生产单元 c 中按照工艺路线 r 加工,那么,多台 m 类型同质机器并行操作工序 s 时的单位加工时间和产品批次 b 在生产单元 c 内完成生产的加工周期分别可以用公式(3.1)和(3.2)表示。

$$t_{s,m}^{p,r} = \frac{t_{s,m}^{p,r}}{\sum_{i=1}^{N_c} U_{m(i)}^c} \quad (3.1)$$

$$F_b^c = \sum_{p=1}^P \sum_{r=1}^{R_p} Y_b^p H_b^r \left[\sum_{s=1}^{S_{p,r}} \sum_{m=1}^M t_{s,m}^{p,r} D_{s,m}^{p,r} + (V_b - 1) \max_s \left\{ \sum_{m=1}^M t_{s,m}^{p,r} D_{s,m}^{p,r} \right\} \right] \quad (3.2)$$

在大多数关于生产单元构建问题的研究中,生产单元构建算法都基于一条重要假设,即相邻工作间的换装时间忽略不计或被归入加工时间。这条假设不仅弱化了产品生产顺序对最大完工时间的影响,还会大大降低衡量生产系统性能时间指标的精确度和问题解决方

案的实用性。由于本论文研究的问题以时间指标来衡量生产单元间工作量均衡程度,为保证结果的正确性,问题中考虑了换装时间。由于产品按照产品批次来组织生产,各批次属于不同的产品类型,因此,各生产单元内相邻的两个生产批次可能属于不同的产品类型。计算换装时间时首先要判断相邻的两个产品批次是否属于不同产品类型,并不是每批产品生产之前都要进行换装。若生产单元 c 中的相邻产品批次 b' 和 b 属于不同产品类型,那么生产批次 b 投入生产前所需的换装时间如式(3.3)所示。

$$S_b^c = t_{b',b} (1 - W_c^{b',b}) \quad (3.3)$$

解决具备完结性的生产单元构建问题的非线性多目标规划数学模型如下所示:

Minimize

$$E = \frac{1}{C} \sum_{c=1}^C \sum_{b=1}^B X_b^c (F_b^c + S_b^c) \quad (3.4)$$

$$D = \sum_{c=1}^C \left[\sum_{b=1}^B X_b^c (F_b^c + S_b^c) - \frac{1}{C} \sum_{c=1}^C \sum_{b=1}^B X_b^c (F_b^c + S_b^c) \right]^2 \quad (3.5)$$

s. t.

$$\sum_{c=1}^C \sum_{r=1}^{R_p} X_b^c H_b^r = 1 \quad \forall b, p, \text{ 且 } Y_b^p = 1 \quad (3.6)$$

$$\sum_{c=1}^C U_{m(i)}^c = 1 \quad \forall m(i) \quad (3.7)$$

$$\sum_{m=1}^M \sum_{i=1}^{N_m} U_{m(i)}^c \leq N \quad \forall c \quad (3.8)$$

$$\sum_{m \in e_{(p,r)}} \left[\frac{\sum_{i=1}^{N_m} U_{m(i)}^c}{N_m} \right] = A_{e_{(p,r)}} \quad \forall b, p, r, c, \text{ 且 } Y_b^p = 1, X_b^c = 1, H_b^r = 1 \quad (3.9)$$

$$\sum_{p=1}^P \sum_{r=1}^{R_p} \sum_{m=1}^M \sum_{i=1}^{N_m} Y_b^p H_b^r D_{s,m}^{p,r} U_{m(i)}^c V_{m(i)}^b = V_b \quad \forall c, b, s, \text{ 且 } X_b^c = 1 \quad (3.10)$$

$$\sum_{b=1}^B \sum_{p=1}^P X_b^c Y_b^p \left[\frac{V_{m(i)}^b}{V_b} \right] \left[t_{b',b} W_c^{b',b} + \sum_{r=1}^{R_p} \sum_{s=1}^{S_{p,r}} H_b^r D_{s,m}^{p,r} t_{s,m}^{p,r} V_{m(i)}^b \right] \leq G \quad \forall c, m(i), \text{ 且 } U_{m(i)}^c = 1 \quad (3.11)$$

模型中的两个目标函数分别是所有生产单元生产总流程时间的均值最小化和各生产单元生产总流程时间的方差最小化。生产单元生产总流程时间的均值最小化旨在缩短所有生产单元的生产总流程时间,从整个生产系统的角度考虑,是为了让所有生产批次能够在尽可能短的时间内完成生产,即缩短最大完工时间。根据第一个目标函数得到的均值,可

以确定整个生产系统中生产单元的平均工作量,然后通过满足第二个目标函数可以降低各生产单元间工作量的差值,使各生产单元的工作量都趋于平均水平,实现生产单元间工作量均衡,消除过载生产单元和空闲生产单元的出现。通过缩短最大完工时间和各生产单元间的工作量差值,可以明显降低产成品库存成本,提高发货速度。

公式(3.6)表示每一个产品批次只能被分配到一个生产单元内按照一条工艺路线组织生产。公式(3.7)确保每台机器只能被分配到一个生产单元内。公式(3.8)限制了每个生产单元可容纳的机器数上限。公式(3.9)规定当产品批次 b 按照工艺路线 r 生产时,负责生产产品批次 b 的生产单元内必须拥有工艺路线 r 内包含的所有类型的机器。也就是说产品批次 b 的所有工序都能够在指定的生产单元内完成,确保生产过程的完结性和生产单元的独立性。公式(3.10)表示当产品批次 b 的某道工序在多台同质机器上平行加工时,每台同质机器上分配的生产量之和必须等于产品批次 b 的总批量。公式(3.11)确保每台机器上分配的工作量之和不得超过机器在给定时间段内的可用生产能力上限。

3.4 基于非支配排序遗传算法-II (NSGA-II) 的混合遗传算法开发

考虑到生产单元构建问题的复杂性和遗传算法解决组合优化问题的有效性,本节给出了一个混合遗传算法对所建立的数学模型进行求解。该混合遗传算法在非支配排序遗传算法-II 的基础上,加入了启发式算法,同时结合了分组遗传算法的理念。遗传算法负责产生初始种群并通过进化过程不断改进种群中个体的质量,其中,杂交算子的设计结合了分组遗传算法的思想,改进后的杂交算子更适合本问题的特征和染色体结构。混合遗传算法中嵌入的启发式算法主要用于对初始种群和杂交后产生的种群中的个体进行可行性校验以及给种群中个体所代表的机器和产品批次分组方案分配工作量。可行性校验通过适当地调整将不满足约束条件的解转化为可行解。由于染色体编码的特殊性,染色体中无法包含产量信息,所以要借助启发式算法来分配,否则无法按照问题的目标函数计算种群中个体的适应值。由于本论文研究的生产单元构建问题是个多目标决策,本节提出的算法利用非支配排序遗传算法-II 求得问题的一组帕累托最优解,该算法流程如图 3-3 所示。

遗传算法是基于种群的算法,它以一组解为对象,通过种群不断进化,最终得到最优解或近似最优解,因此,在算法开始时,首先要给出一组解作为进化过程的起点。算法开始,首先随机产生 N 个个体,这 N 个个体组成初始种群 P_0 。用校验启发式算法判断初始种群中个体所代表的解的可行性,并将不满足约束条件的个体调整为可行解。接着,根据个体所代表的解中机器和产品批次的分组情况,利用工作量分配启发式算法将各产品批次的生产任务分配的具体的机器上,然后,根据上节数学模型中的目标函数计算每个个体的适应值。根据适应值,用快速非支配排序方法(The fast non-dominated sorting approach)将种群中所有个体划分为不同的帕累托边界,给属于同一条边界的个体分配相同的等级。接下来,结合个体的非支配等级和拥挤距离,使用基于拥挤距离的锦标赛选择法挑选出用于产生下一代的父代。利用杂交算子操作选择的父代,产生个体数为 N 的子代 Q_0 ,然后,

将 P_0 和 Q_0 整合为一个个体数为 $2N$ 的种群 R_0 。对种群 R_0 使用快速非支配排序方法，排序后排在最前面的个体代表较好的解，从序列中选择前 N 个个体作为新的父代 P_1 。对 P_1 进行非支配排序，使用选择算子和杂交算子产生新一代 P_2 。就这样，上面的流程不断重复，种群一代代的进化，直至达到给定的最大遗传代数，方可停止。

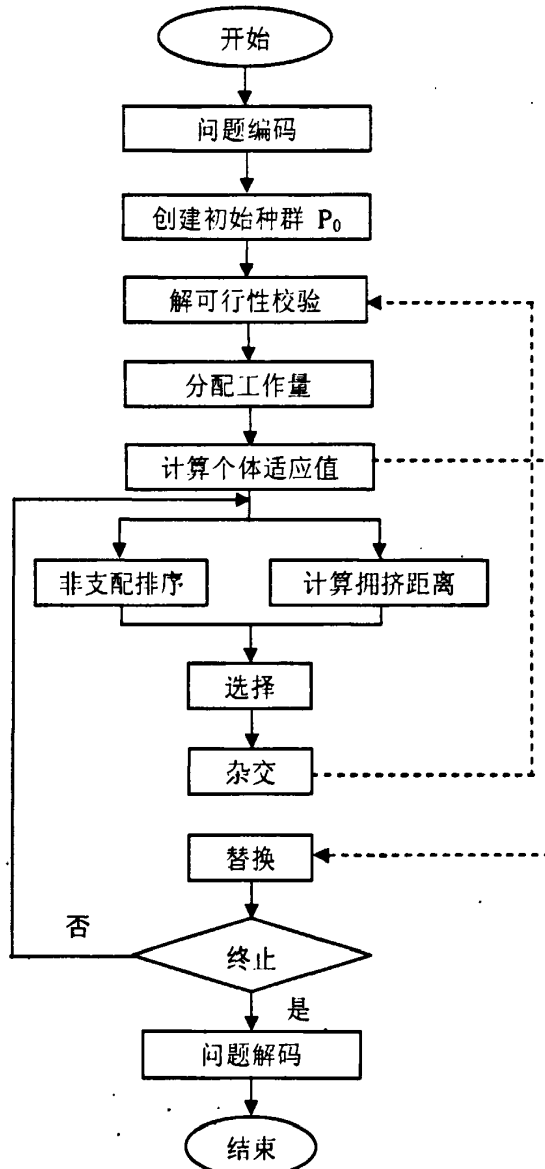


图 3-3 混合遗传算法流程图

Fig. 3-3 The flow chart of the proposed hybrid genetic algorithm

3.4.1 编码规则

遗传算法中问题的解由染色体表示，染色体由一串基因组成。每个基因都是一个变量，这个变量的值可以是二进制数、实数和字母。本算法中使用的染色体中同时包含了机器、产品批次和工艺路线的相关信息，其表现形式是对 Joines 等人^[92]给出的染色体形式的扩展。本算法中使用的染色体结构如图 3-4 所示，染色体长度为 $M+2B$ ，其中， M 为机器的

数量, B 为产品批次的数量, 因此, 染色体的长度随问题的规模而变, 问题中考虑的机器和产品批次数量越大, 染色体就会越长。图中整个染色体被竖线划分为三个部分, 第一部分的基因个数等于问题中考虑的机器数量, 后两个部分的基因个数等于问题中考虑的产品批次数量。前两个部分的基因值是[1,C]范围内的随机数, C 是拟建立生产单元个数, 基因值表示机器或产品批次分配到的生产单元编号。第三部分的基因值表示每个产品批次选定的工艺路线编号, 基因值的取值范围随各产品批次所属产品类型的工艺路线总数变化。

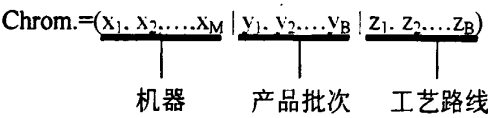


图 3-4 染色体结构

Fig. 3-4 The structure of chromosome

为了更好地解释本算法中染色体的编码规则, 现以一个将 9 台机器和 5 个产品批次划归为 3 个生产单元的问题为例, chromosome1 代表问题的一个解。若 chromosome1=(1 2 1 3 1 3 2 1 3 2 3 1 3 1 1 2 2 1 1), 则该染色体所代表的解的含义是编号为 1,3,5,8 的机器组成生产单元 1, 负责生产产品批次 3 和 5; 编号为 2 和 7 的机器组成生产单元 2, 负责生产产品批次 1; 编号为 4,6,9 的机器组成生产单元 3, 负责生产产品批次 2 和 4。其中产品批次 1 按照其所属产品类型的工艺路线 1 组织生产, 相似的, 产品批次 3 按照其所属产品类型的工艺路线 2 组织生产。

上文提到的染色体结构是初始种群中随机产生的个体所采用的结构。考虑到生产单元构建问题的特殊性, 机器和产品批次划分后得到的分组结果中每一个生产单元都是一个完整的模块, 因此, 为了保证各生产单元的完整性不被破坏, 在使用杂交算子产生下一代之前, 需将父代种群中的个体转化为另一种结构。新的染色体和原有染色体所代表的解是一样的, 只不过在新染色体结构中机器和产品批次按照生产单元分组情况排列, 这样有利于在杂交过程中以生产单元为单位交换基因, 避免产生大量不可行解。参照 Gonçalves Filho 和 Tiberti^[93]提出的染色体编码方法, 本算法杂交过程中使用的染色体结构如图 3-5 所示。染色体前两部分将机器和产品批次按照原染色体中解的含义组合为若干生产单元, 染色体中的基因值分别代表机器编号和产品批次编号, 这样可以明显看出每个生产单元包含的机器和负责生产的产品批次。染色体第三部分依然表示各产品批次的工艺路线, 这部分的基因值没有改变, 只是根据第二部分中产品批次的顺序进行重新排序, 以确保产品批次和工艺路线的匹配。与第一种染色体结构相比, 对第二种染色体结构进行杂交后产生的子代依然满足对生产单元可容纳机器数的约束, 也不会出现子代的生产单元数少于父代的情况。

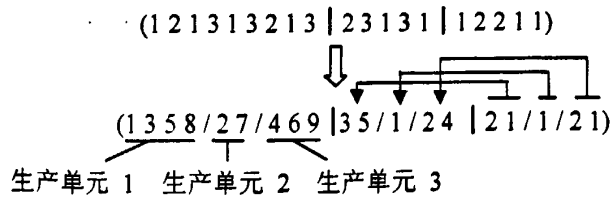


图 3-5 杂交过程中使用的染色体结构

Fig. 3-5 The structure of chromosome used for crossover

3.4.2 创建初始种群

遗传算法中，一组染色体的集合被称为一个种群，在一个给定时期内出现的种群被称为一代。初始种群被视为进化过程的起点，仅在遗传算法开始的时候创建一次。通过在初始种群中选择出优秀的个体作为父代，并利用遗传算子产生子代，进化过程就开始向前推进。创建初始种群时要根据事先规定的种群规模产生相应数量的个体，最常见的创建方法是随机产生个体和借助启发式算法。很多遗传算法中都使用随机产生个体的方法，但是考虑到生产单元构建问题的特征和数学模型中的严密约束，本论文提出的遗传算法使用特定的程序来随机产生初始种群，以避免个体所代表的解中出现没有被分配到任何机器和产品批次的空生产单元。借鉴 Gonçalves Filho 和 Tiberti^[93] 给出的初始种群创建程序，本算法依据以下步骤产生初始种群的所有个体：

- (1) 针对每个生产单元 c_i (i 从 1 到 C)，从机器群和各产品批次中分别随机选择一台机器和一个产品批次，并给选定的机器和产品批次所对应的基因赋值为 i ；
- (2) 针对其余未分配的 $(M-C)$ 台机器和 $(M-C)$ 个产品批次，每次选择一台机器或一个产品批次，并随机产生一个生产单元编号，将该编号赋值给所选机器或产品批次对应的基因，此过程不断重复直至所有机器和产品批次都被分配给生产单元；
- (3) 针对每个产品批次，从该批次所属产品类型的工艺路线集合中任选一条工艺路线，并将工艺路线编号赋给对应的基因；
- (4) 前面三个步骤得到的结果分别构成染色体的三部分，将染色体写为第一种结构并放入初始种群中；
- (5) 重复步骤(1)-(4)，直至初始种群中的个体数量满足所给种群规模的要求。

3.4.3 可行性校验

初始种群中随机产生的个体和杂交过程产生的新个体通常不满足具体问题中的约束条件，因此，要使用可行性校验程序判断这些个体所代表的解的可行性，并通过适当的调整将不满足约束条件的个体转换为可行解。本节给出的校验程序主要检查个体所代表的解是否满足生产单元容纳能力上限约束（公式(3.8)）和生产过程完结性约束（公式(3.9)），校验程序如下：

- (1) 将每个个体转化为图 3-5 所示的染色体结构；
- (2) 随机选择一个个体 i ；

(3) 选择个体 i 中的一个产品批次 b , 若产品批次 b 被分配到生产单元 c^* , 查看该生产单元中所具备的机器类型是否满足产品批次 b 选定工艺路线下各工序的需求, 如果不满足, 执行步骤(4);

(4) 根据产品批次 b 选定工艺路线下各工序需要使用的机器类型, 确定生产单元 c^* 中缺少的机器类型。对于缺少的机器类型 m' , 首先找出除 c^* 外拥有该类型机器的生产单元, 然后确定各生产单元中该类型同质机器的数量, 从拥有最多同质机器的生产单元 c' 中抽出一台属于类型 m' 的机器并放入生产单元 c^* ; 若各生产单元中该类型同质机器的数量相同, 则从任意生产单元中抽出一台属于类型 m' 的机器放入生产单元 c^* ;

(5) 重复步骤(3)和(4), 直到个体 i 中所有的产品批次都满足公式(3.9);

(6) 检查每个生产单元中分配的机器数量是否超过生产单元容纳能力上限, 若超过, 执行步骤(7), 否则, 停止校验程序;

(7) 对任意不满足生产单元容纳能力约束的生产单元 c , 找出该生产单元中包含同质机器数最多的机器类型 m ;

(8) 将属于 m 类型的任一同质机器 m^* 重新分配给拥有 m 类型机器且同质机器数最少的生产单元。若其他生产单元都不包含 m 类型机器, 则将 m^* 重新分配给拥有各类机器总数最少的生产单元。重复这一分配过程, 直到生产单元 c 满足生产单元容纳能力约束;

(9) 重复步骤(7)和(8), 直到所有生产单元都满足公式(3.8);

(10) 如果种群中全部个体都已校验完毕, 就停止校验程序, 否则, 执行步骤(2)。

3.4.4 计算适应值

初始种群一旦建立, 就要根据数学模型中的目标函数方程和约束条件来计算每个个体的适应值。适应值方程将染色体代表的个体转化为解空间里的一个实际值, 通过这个值来确定用于产生下一代个体的父染色体集。但是由于受到染色体结构的限制, 从染色体中只能得出机器和产品批次的分组情况, 以及各产品批次的工艺路线选择情况, 若要求得目标函数值, 还必须得到各产品批次的工作量具体分配方案。在每个个体已经得到的解的基础上, 可利用本节给出的启发式算法来分配工作量, 算法步骤如下:

(1) 根据种群中个体的染色体反映出的信息, 为每个生产单元建立一个机器类型集 M_c 和产品批次集 B_c , 并计算每类机器的同质机器数 N_c^m ;

(2) 选择一个没有分配工作量的生产单元 c , 如果所有生产单元都已分配完毕, 则停止;

(3) 从生产单元 c 的产品批次集 B_c 中选择一个尚未被分配的产品批次 b , 如果所有产品批次都已分配完毕, 则执行步骤(2);

(4) 根据染色体中产品批次 b 对应的工艺路线 r , 确定产品批次 b 所属的产品类型按照工艺路线 r 生产时用到的所有机器类型;

(5) 计算每道工序所对应机器上的最佳平均工作量 w_b^* , 若产品批次 b 的工序 s 在机

器类型 m 上加工, 则每台同质机器上分配的最佳平均工作量为 $w_b^* = \left\lceil \frac{V_b}{N_c^m} \right\rceil$;

(6) 从生产单元 c 中任选 $(N_c^m - 1)$ 台属于 m 类型的同质机器, 并给每台机器上分配 w_b^* 个单位的工作量, 接着将剩余的 $[V_b - (N_c^m - 1)w_b^*]$ 个单位的工作量分配给最后一台 m 类机器。

(7) 重复步骤(5)和(6), 直至产品批次 b 的工艺路线 r 中所有工序的工作量分配完毕;

(8) 重复步骤(2)-(6), 直至所有产品批次的工作量分配完毕。

本论文研究的生产单元构建问题是一个受约束的最优化问题, 所以仅当种群中个体所代表的解满足所有约束条件时, 这个解才是可行解。但是用遗传算法来解决这种受到严格约束的问题时, 种群中相当大一部分解都可能是不可行解^[94]。遗传算法中用于处理不可行个体的方法很多, 包括拒绝、惩罚和修补等^[79]。考虑到本论文提出的数学模型的复杂性, 对不满足六个约束条件的情况分别采用不同的方法来处理。从使用的染色体结构可以看出, 每台机器和每个产品批次分别占一个基因位, 因此它们分别只能分配给一个生产单元, 且每个产品批次只能分配一条工艺路线, 因此, 种群中的个体都能满足前两个约束条件, 即公式(3.6)和(3.7)。上文中给出的可行性校验程序能够确保初始种群中的个体和杂交后产生的子代中的个体满足公式(3.8)和(3.9)。对于公式(3.10), 从工作量分配启发式算法给出的分配方法可以看出, 所有个体的工作量分配结果都能满足该项约束。由于最后一个约束条件(公式(3.11))在很大程度上影响着目标函数计算值的准确性, 且不满足该项约束的解很难通过简单调整来处理, 因此, 本论文使用罚函数来处理不满足该项约束的个体。

使用罚函数通过将约束条件与目标函数合并, 可以把有约束的问题转化为不含约束的问题。在目标函数值中加入惩罚值能够降低代表不可行解的个体的适应值, 进而降低该个体被选作父代产生下一代的几率。本算法中使用的罚函数是在 Gen 和 Cheng^[95]提出的罚函数基础上简化得到的。这种罚函数在每一代根据实际情况适当地调整惩罚率, 这样不但可以避免出现过度惩罚现象, 还能在保留有用信息的同时对不可行解施加惩罚压力。

Gen 和 Cheng^[95]提出的罚函数针对最大化问题同时处理多个约束条件, 本算法中罚方法只用来处理不满足公式(3.11)的解, 固将原函数简化为针对最小化问题的处理一个约束条件的形式。对于最小化问题来说, 适应值越小的个体所代表的解最优, 那么以相乘形式给出的罚函数 $p(x)$ 应满足:

$$\begin{cases} p(x) = 1, \text{ 当 } x \text{ 是可行个体时;} \\ p(x) > 1, \text{ 其他。} \end{cases} \quad (3.12)$$

若 x 是种群 $P(t)$ 中的一个个体, 则简化后的罚函数应为:

$$p(x) = 1 + \left(\frac{\Delta b(x)}{\Delta b_{\max}} \right)^k, \quad (3.13)$$

$$\Delta b(x) = \max \{0, g(x) - G\}, \quad (3.14)$$

$$\Delta b^{\max} = \max \{ \varepsilon, \Delta b(x) | x \in P(t) \}, \quad (3.15)$$

其中 $\Delta b(x)$ 表示个体 x 所代表的解对公式(3.11)的违背值, Δb^{\max} 是当前种群中所有个体对公式(3.11)违背值的最大值, k 是由用户定义的用来调节惩罚力度的参数, ε 是一个非常小的正数, 用来避免公式出现被零除错误。加入罚函数后, 用来衡量种群中个体质量的适应值函数为:

$$\begin{cases} eval^1(x) = f_1(x)p(x), \\ eval^2(x) = f_2(x)p(x). \end{cases} \quad (3.16)$$

3.4.5 非支配排序

对于多目标优化问题, 最优解需要同时满足所有目标函数。大多数情况下, 多目标优化问题被转化为单目标问题求解, 最常见的方法就是对每个目标函数赋予权重后转变成各目标函数相加的线性形式。但这样转化后会导致搜索过程中无法发现解空间中一些可能存在最优解的区域, 因此有可能丢失一些最优解。本文使用非支配排序遗传算法-II (NSGA-II) 来处理多目标优化问题, 利用这种有效方法能够快速得到问题的帕累托最优解集合。

首先, 将整个种群划分为若干帕累托边界, 给属于同一条边界的个体分配一个级别, 这个级别代表它们的非支配等级。对拥有 N 个个体的初始种群 P_0 进行非支配排序, 利用 Deb 等人^[65]提出的快速非支配排序方法将所有个体划分为不同的帕累托边界。开始排序之前, 首先根据每个个体的染色体表示计算两个参数值, n_p 和 S_p , n_p 表示支配个体 p 的解的个数, S_p 是被个体 p 支配的解的集合。对于第一条帕累托边界中的个体来说, 种群中没有任何个体能够支配它们, 因此这些个体的 n_p 值为 0。对每一个 $n_p = 0$ 的个体 p , 将 S_p 中每个成员 q 的 n_q 值减 1。接着将那些 n_q 值变为 0 的成员 q 放入一个单独的列表 Q 中, 列表 Q 中的所有成员构成第二条帕累托边界。将列表 Q 中的每个成员 u 的 n_u 值减 1, 确定第三条帕累托边界。以上过程不断重复, 直至得到所有帕累托边界。由上可知, 等级 1 中的个体不被任何个体支配, 等级 2 中的个体只被等级 1 中的个体支配, 依此类推。快速非支配排序过程的伪代码如图 3-6 所示。

属于同一条帕累托边界的个体最好是均匀分布的, 也就是说, 个体间维持较好的多样性能够避免算法结果快速收敛到局部最优解。拥挤距离是衡量种群中个体与其相邻个体接近程度的指标, 拥挤距离越大, 种群的多样性水平越高^[96]。对于每一条边界对应的帕累托解集, 首先将个体分别按照两个目标函数对应的适应度值进行升序排序, 接着给排序后处于两个队列边界上的个体(适应度值最小和最大的个体)赋予正无穷大距离, 以确保这些个体能够存活到下一代。然后, 利用公式(3.17)计算其他个体的拥挤距离,

$$C_j = \sum_{i=1}^{N_{\text{obj}}} \frac{F_i^{j+1} - F_i^{j-1}}{(F_i^j)_{\max} - (F_i^j)_{\min}}, \quad (3.17)$$

其中, N_{obj} 是目标函数个数, F_i 是一条帕累托边界中第 j 个个体的第 i 个目标函数值。

```

for each  $p \in P$ 
     $S_p = \phi$ 
     $n_p = 0$ 
    for each  $q \in P$ 
        if  $(p \prec q)$  then
             $S_p = S_p \cup \{q\}$ 
            如果  $p$  支配  $q$ 
            将  $q$  添加至被  $p$  支配的解集中
        else if  $(q \prec p)$  then
             $n_p = n_p + 1$ 
            增加  $p$  的支配解计数器值
    if  $n_p = 0$  then
         $p_{rank} = 1$ 
         $F_1 = F_1 \cup \{p\}$ 
         $p$  属于第一条帕累托边界
     $i = 1$ 
    while  $F_i \neq \phi$ 
         $Q = \phi$ 
        储存下一条帕累托边界的成员
        for each  $p \in F_i$ 
            for each  $q \in S_p$ 
                 $n_q = n_q + 1$ 
                if  $n_q = 0$  then
                     $q_{rank} = i + 1$ 
                     $Q = Q \cup \{q\}$ 
                     $q$  属于下一条帕累托边界
             $i = i + 1$ 
         $F_i = Q$ 
    
```

图 3-6 非支配排序程序

Fig. 3-6 The procedure of non-dominated sorting. Source: Deb et al. (2002)

3.4.6 选择算子

通常情况下, 每个种群中适应度最高的个体会被选作繁衍下一代的候选人, 因为它们能把较好的基因传递给下一代, 从而改善后续各代的个体质量, 达到朝着更好的方向不断进化的目的^[92]。在对上一代 P_t 进行非支配排序后, 使用基于拥挤距离的锦标赛选择法挑选种群中优秀的个体作为产生下一代的父代。在选择父代时主要依据个体的非支配等级和拥挤距离, 对于任意两个随机抽选的个体, 将非支配等级级别较低的那个加入父代序列中。若两个个体属于同一个非支配等级, 处于相对不拥挤区域的个体, 即拥挤距离值较大的个体, 将被选作父代, 因为这样能够维持种群的多样性。

3.4.7 杂交算子

尽管每一代中的个体数量相对整个解空间来说非常小, 但是利用杂交和变异能够深入探索许多未曾被发现, 却有可能找到最优解的区域, 这样更利于找到更好的解。先前研究

曾指出,对于用来解决分组问题的遗传算法来说,变异算子不会对改善解的质量和算法性能起到太大作用,因此一些研究中都没有考虑变异算子,本文提出的算法也是如此^[97]。为了能够提高遗传算法解决分组问题的效率,先前研究中提出了许多专门用于解决分组问题的杂交算子,例如单点杂交算子、两点杂交算子、基于生产单元的交流算子和基于生产单元的两点算子。为了适应本论文算法的染色体结构,基于生产单元的两点杂交算子被拓展为三点杂交算子。

杂交算子能够将双亲染色体的特征相结合,并产生子代。用于产生下一代的双亲是使用基于拥挤距离的锦标赛选择法从上一代中随机挑选出来的。双亲杂交前,首先要从 1 到 $k-1$ (k 是生产单元总数)中随机选择三个杂交点,三个杂交点分别对应双亲染色体中机器、产品批次和工艺路线这三个部分。接着,分别交换双亲染色体的三个部分中位于杂交点右边的各组基因,产生两个新的个体。需要特别说明的是,为了避免出现杂交后子代染色体中产品批次和工艺路线不匹配的情况,产品批次和工艺路线这两部分的杂交点必须相同。

现以一个将 10 台机器和 5 个产品批次划分为三个生产单元的问题为例,详细阐述个体杂交过程。图 3-7 给出了包含该问题相关信息的机器-产品批次关联矩阵,元素 a_{ij} 的值 x 表示在指定工艺路线下机器 i 加工产品批次 b 的第 x 道工序。从种群中挑选两个个体 P1 和 P2 作为双亲:

$P1=(3\ 1\ 1\ 2\ 3\ 1\ 2\ 1\ 3\ 2\ 2\ 1\ 3\ 2\ 3\ 3\ 1\ 2\ 1\ 1)=(2\ 3\ 6\ 8/4\ 7\ 10/1\ 5\ 9|2/1\ 4/3\ 5|1/3\ 1/2\ 1)$

$P2=(1\ 3\ 2\ 1\ 1\ 3\ 3\ 2\ 3\ 1\ 1\ 3\ 1\ 2\ 3\ 1\ 2\ 1\ 2\ 2)=(1\ 4\ 5\ 10/3\ 8/2\ 6\ 7\ 9|1\ 3/4/2\ 5|1\ 1/2/2\ 2)$

Product batch			Pb 1			Pb 2		Pb 3			Pb 4		Pb 5	
Process routing			1	2	3	1	2	1	2	3	1	2	1	2
Column number			1	2	3	4	5	6	7	8	9	10	11	12
Machine	Type I	1	1	1				3	1	4				
	Type II	2				2	2			3		1		
		3				2	2			3		1		
	Type III	4	2		2			2	2		1		1	
		5	2		2			2	2		1		1	
	Type IV	6	3	2		1	1			2				2
	Type V	7		3	3	3					3	2	2	1
		8		3	3	3					3	2	2	1
		9		3	3	3					3	2	2	1
	Type VI	10	4		1			1		1	2			

图 3-7 机器-产品批次关联矩阵

Fig. 3-7 The machine-product batch (MP) incidence matrix

假设每个部分的杂交点为 2,1,1,则交换杂交点右侧的基因后产生的两个子染色体 C1' 和 C2' 如图 3-8 所示。

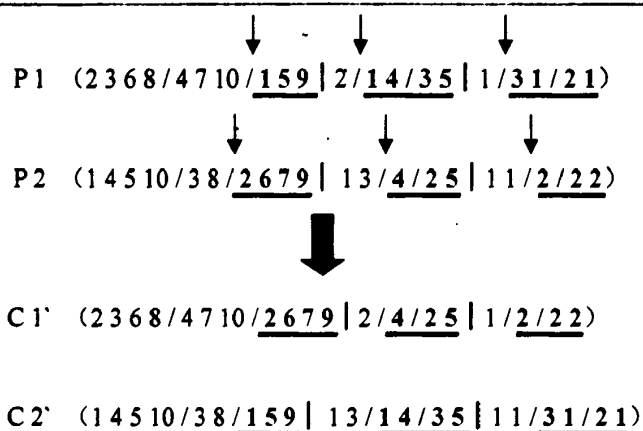


图 3-8 杂交算子操作过程

Fig. 3-8 The numerical illustration of the work of crossover operator

杂交过程产生的新个体中有可能缺少某些必要的基因或包含一些冗余的基因。以子染色体 C1' 为例，基因串中缺少机器 1 和 5，产品批次 1 和 3，同时机器 2,6,7 和产品批次 2 出现冗余。由于这个子染色体所代表的解中一些机器和产品批次没有被分配给任何生产单元，另一些机器和产品批次却被多次分配给不同的生产单元，所以这个染色体所代表的解是不可行解。针对这种情况，现给出一个基于本论文所研究问题的调整方案来修正不符合条件的子染色体，使其所代表的解成为可行解。

首先，删去那些在子染色体中出现两次的机器和产品批次。对于出现两次的机器和产品批次，删去杂交点左边的那个副本，保留通过交换得到的那个副本。在删去产品批次时，与其相对应的工艺路线也应一同删去。这种处理方法使染色体中的新成员代替旧成员^[78]。将图 3-8 中 C1' 和 C2' 中重复元素删去后，得到图 3-9 中的 C1'' 和 C2''。但是这样处理后，却出现了一些不包含任何机器和产品批次的空生产单元（图 3-9 中 C1'' 的生产单元 1）。如果杂交点左侧存在空生产单元，找出染色体中缺少的机器和产品批次，根据空生产单元在染色体中所处的区域，从缺少的机器或产品批次中任选一个分配给空生产单元。分配缺少的产品批次时，其工艺路线必须和原所属双亲中的工艺路线相同。若将缺少的机器和产品批次全部分配给空生产单元后仍然存在空生产单元，则从杂交点左侧的非空生产单元中任选一个机器或产品批次分配给空生产单元。该过程一直重复直至子染色体中不存在空生产单元。将图 3-9 中的空生产单元填补后得到 C1''' 和 C2'''。若子染色体中没有空生产单元或在填补了所有空生产单元后依旧缺少机器或产品批次，则按照下文给出的方法处理。

基于 Brown 和 Sumichrast^[97] 给出的替换启发式算法的思想，本算法根据以下步骤来分配子染色体中缺少的机器或产品批次，分配依据为机器的加工能力和产品批次指定工艺路线的加工需求。

步骤 1：将所有缺少的机器和产品批次分别放入两个列表中，列表中元素以放入先后顺序为序。

步骤 2：选出机器列表中的第一台机器 m，找到机器 m 在机器-产品批次关联矩阵中对应的行 i。根据染色体中所有产品批次的工艺路线，在机器-产品批次关联矩阵中标出对

应的列。检查这些列与 i 行的交点，找出交点上的元素值为任意整数值的列，并记录这些列对应的产品批次和工艺路线信息。

步骤 3：查看染色体中所有生产单元的产品批次和工艺路线信息，并与步骤 2 得到的结果相对比，选择与步骤 2 得到的产品批次和工艺路线信息匹配度最高的生产单元 c ，并将机器 m 分配给生产单元 c ；若同时存在多个匹配度高的生产单元，则将机器 m 分配给机器总数最少的生产单元；若这些匹配度高的生产单元的机器总数相同，则任选一个生产单元接受机器 m 。

步骤 4：若仍然缺少机器，执行步骤 2，否则，执行步骤 5。

步骤 5：选出产品批次列表中的第一个产品批次 b ，根据其原所属双亲中的工艺路线 r 找到产品批次 b 在机器-产品批次关联矩阵中对应的列 j 。检查机器-产品批次关联矩阵中各行与 j 列的交点，找出交点上的元素值为任意整数值的行，并记录这些行对应的机器编号。

步骤 6：查看产品批次 b 的工艺路线 r 下各工序所需的机器类型，并与步骤 5 得到的结果相对比，选择与步骤 5 得到的机器信息匹配度最高的生产单元 c ，并将产品批次 b 分配给生产单元 c ；若同时存在多个匹配度高的生产单元，则将产品批次 b 分配给产品批次总数最少的生产单元；若这些匹配度高的生产单元的产品批次总数相同，则任选一个生产单元接受产品批次 b 。

步骤 7：若仍然缺少产品批次，执行步骤 5，否则，停止程序。

根据以上步骤，调整算例中的染色体。图 3-9 中的空白单元填补后得到 $C1''$ 和 $C2''$ ，现在要将其他缺失的机器添加到染色体中。以 $C1''$ 中缺少的机器 1 为例，从机器-产品批次关联矩阵中可以看出，加工过程中需要使用机器 1 的产品批次包括：产品批次 1（工艺路线 1,2）和产品批次 3（工艺路线 1,2,3）。由于生产单元 1 包含在工艺路线 2 下加工的产品批次 3，其他生产单元的产品批次在加工过程中都不需要使用机器 1，因此，将机器 1 分配到生产单元 1。用相同的办法处理其他缺失机器后，得到杂交过程的最终结果 $C1^*$ 和 $C2^*$ ，杂交结果的整个调整过程如图 3-9 所示。整个杂交过程结束后，使用可行性校验程序再次处理杂交结果，以保证最终得到的子代中的个体所代表的解都是问题的可行解。

3.4.8 替换规则

种群 P_t 中选择出的双亲染色体完成杂交过程后，产生一个个体数为 N 的子代 Q_t 。将 P_t 和 Q_t 组合成一个个体数为 $2N$ 的新种群 R_t ，并对 R_t 进行非支配排序。根据个体的非支配等级，将 R_t 中的所有个体按非支配等级由高到低排序，从中选出 N 个最好的个体组成新的父代 P_{t+1} 。挑选优秀个体组成新的父代时遵循如下规则：首先计算出每条帕累托边界上的个体数目，从等级为 1 的边界开始，若该边界上个体的数量不大于新的父代需要的个体数，则将该条边界上的所有个体都放入新的父代中。若边界上个体的数量大于新的父代需要的个体数，则将该条边界上的个体按照拥挤距离排序，根据新的父代所需的数量从拥

挤距离值最大的个体中挑选出个体放入新的父代。这个挑选过程从非支配等级最高的边界开始，不断向后面的边界推进，直到新的父代 P_{t+1} 的种群规模达到事先给定的数量 N 。由于 R_t 中包含了上一代和当代种群中的全部个体，上述替换规则能够确保种群中的精英个体都保留在新一代 P_{t+1} 中^[85]。新一代 P_{t+1} 产生后，继续从中选择双亲染色体，通过杂交产生下一代，整个进化过程就这样不断向前推进。

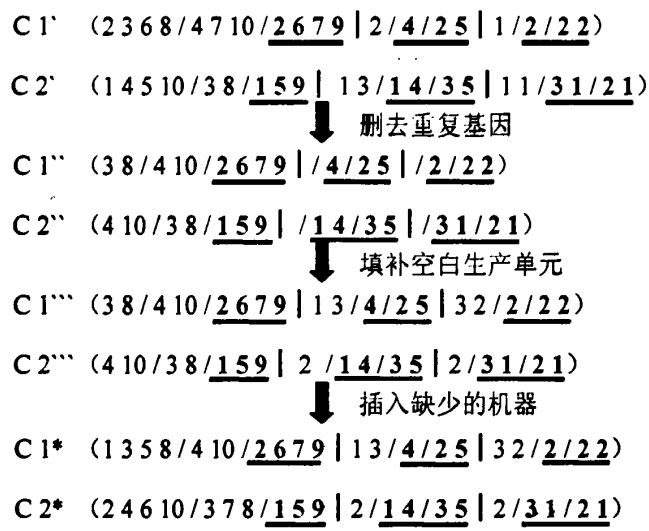


图 3-9 子代染色体调整过程

Fig. 3-9 The adjustment procedure for offsprings generated by crossover operator

3.5 算例分析

本节使用一组随机产生的数据验证上节给出的基于非支配排序遗传算法-II (NSGA-II) 的混合遗传算法的性能，所给算法使用 MATLAB 7.0 编码，并在一台使用 Windows XP 系统、2.66GHZ Pentium 处理器的电脑上进行运算。

3.5.1 算例描述

表 3-1 给出了算例中使用的机器-产品关联矩阵，矩阵中的整数值代表工序编号，括号里面的值表示加工时间。表 3-2 给出了每个产品批次的批量和其所属的产品类型。表 3-3 和 3-4 分别给出了各类产品间的换装时间和遗传算法的基本参数。

表 3-1 机器-产品批次关联矩阵

Table 3-1 The machine-product batch (MP) incidence matrix

机器类型 \ 产品类型	P1		P2	P3			P4	
	R1	R2	R1	R1	R2	R3	R1	R2
I 类(M1-M5)	2(24.02)			1(17.96)		3(15.05)	3(20.13)	
II 类(M6-M12)		1(25.79)	3(20.21)			1(17.64)		2(23.4)
III 类(M13-15)			1(19.45)		2(15.56)			
IV 类(M16-20)	4(23.94)		4(18.03)	2(19.04)				
V 类(M21-25)		2(23.21)				2(16.93)	1(19.18)	

续表 3-1

VI 类(M26-M30)	3(21.12)			3(16.5)		4(14.4)		1(24.98)
VII 类(M31-M37)	1(26.64)	3(26.85)	2(20.58)		1(17.96)	5(19.61)	2(21.48)	

表 3-2 产品批次信息

Table 3-2 The information of product batches

批次编号	B1	B2	B3	B4	B5	B6	B7
产品类型	4	2	1	3	2	4	1
批量	251	121	189	167	241	217	162

表 3-3 换装时间表

Table 3-3 The list of setup time

	P1	P2	P3	P4
P1	0	8.5	15	11.2
P2	8.5	0	10.6	9.5
P3	15	10.6	0	13.8
P4	11.2	9.5	13.8	0

表 3-4 遗传算法参数表

Table 3-4 The parameters of genetic algorithm

参数	值
染色体长度	51
种群规模	40
遗传代数	100-200
ϵ	1
k	2
生产单元可容纳机器数量	随生产单元数变化
机器可用加工能力	2000 (min)

本论文研究的生产单元构建问题中,生产单元的数量不是已知的,而是一个决策变量。最佳的生产单元数量要根据机器数量、加工能力和产品加工需求来确定,因此,算例要在不同的生产单元数量下多次运算,根据结果判断最佳生产单元数量。由于机器数量固定,所以生产单元可容纳机器数量会随生产单元数的变化而变化。可建生产单元的最大数量应该小于等于机器总数和产品批次总数,因此,本算例中可建生产单元数量的范围是[2,7]。由于遗传算法的初始种群是随机产生的,为了保证解的质量,每一种生产单元数量的情况下程序重复运算 30 次,然后从 30 次的结果中选择出最优解。

3.5.2 算例结果分析

表 3-5 和图 3-10 展示了生产单元数量对 Seru 生产系统性能的影响。从图中可以看出,随着生产单元数量的增加,非支配解集中各非支配解第一个目标函数值的均值递减,第二

个目标函数值的均值呈增加趋势。也就是说，随着生产单元数量的增加，分配到每个生产单元的工作量递减，但各生产单元间的工作量不均衡程度会逐渐增加。生产单元数量的变化会直接影响到每个生产单元内可容纳机器数量上限的变化，本文以构建四个生产单元的情况为例，分析了生产单元可容纳机器数量上限值对系统性能的影响，结果如图 3-11 所示。从图中可以看出，生产单元可容纳机器数的变化对第一个目标函数值的影响较弱，然而不同情况下非支配解集中各非支配解第二个目标函数值的均值波动非常大。可见，在构建生产单元时，不但要选择合适的生产单元数量，还要确定各生产单元可容纳机器数量上限的最佳值。在生产单元数量和生产单元可容纳机器数上限取值不同的情况下得到的解都具有各自的优势，所以企业可以根据自身的侧重点选择合适的生产单元构建方案。

表 3-5 生产单元数量对系统性能的影响

Table 3-5 The impact of cell number on algorithm performance

生产单元数量	各非支配解第一个目标函数 E 的均值	各非支配解第二个目标函数 D 的均值
2	4974.29	12233073.73
3	5801.96	1125151.26
4	5274.72	1136043.21
5	5068.15	947947.34
6	4744.17	2592045.14
7	3730.35	5955350.99

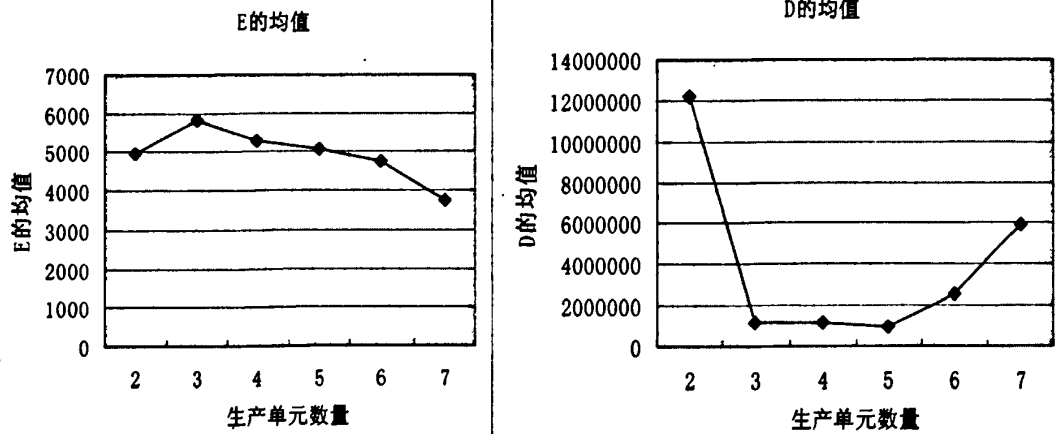


图 3-10 生产单元数量对系统性能的影响

Fig. 3-10 The impact of cell number on algorithm performance

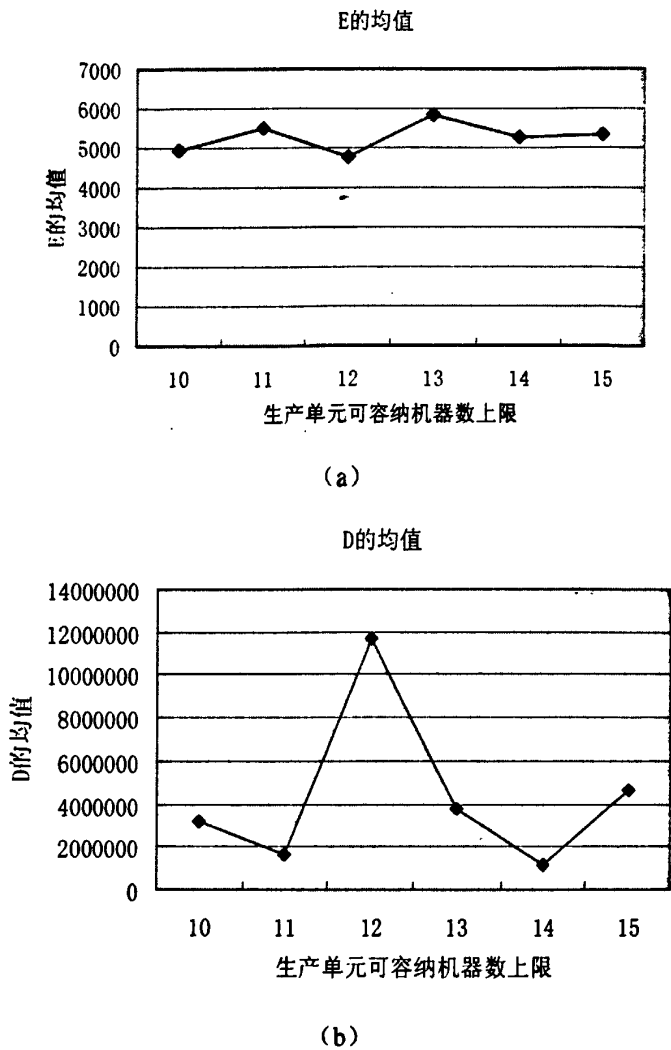


图 3-11 生产单元可容纳机器数上限对系统性能的影响

Fig. 3-11 The impact of cell size on algorithm performance

从表 3-5 中的数据可以看出，构建五个生产单元时得到的非支配解集中解的各目标函数值的均值相对较小，故本算例的最佳生产单元数量为 5。构建五个生产单元时得到的非支配解集中只包含 1 个非支配解，显然这个解是这种情况下的最优解，解的目标函数值分别为 5068.15 和 947947.34，解所对应的工作量分配方案如表 3-6 所示。可见，解中的生产单元都是独立的，产品都能够在一个生产单元内完成生产，没有出现生产单元间物料移动。图 3-12 描述了遗传算法进化过程中每一代的最优解的目标函数的均值的收敛过程，显然随着遗传代数的增加，解的质量在不断提高。图 3-13 展示了建立五个生产单元时的初始种群、终止种群和终止代的非支配解集。通过计算结果可以看出，本文提出的基于非支配排序遗传算法-II 的混合遗传算法能够有效地找到一组非支配解，本文研究的具备完结性的生产单元构建问题得到了很好的解决。

表 3-6 非支配解对应的分块对角矩阵

Table 3-6 The block diagonalized matrix of the non-dominated solution

机器 编号	批次（工艺路线）						
	B4(2)	B6(2)	B7(2)	B5(1)	B1(2)	B2(1)	B3(2)
M9		108(23.4)					
M12		109(23.4)					
M14	167(15.56)						
M18							
M28		109(24.98)					
M29		108(24.98)					
M31	84(17.96)						
M37	83(17.96)						
M1							
M2							
M4							
M10			162(25.79)				
M25			162(23.21)				
M36			162(26.85)				
M5							
M6				241(20.21)			
M13				241(19.45)			
M16				81(18.03)			
M19				81(18.03)			
M20				79(18.03)			
M22							
M35				241(20.58)			
M7					126(23.4)	61(20.21)	
M8					125(23.4)	60(20.21)	
M15						121(19.45)	
M17						121(18.03)	
M23							
M27					125(24.98)		
M30					126(24.98)		
M34						121(20.58)	
M3							

续表 3-6

M11							189(25.79)
M21							94(23.21)
M24							95(23.21)
M26							
M32							95(26.85)
M33							94(26.85)

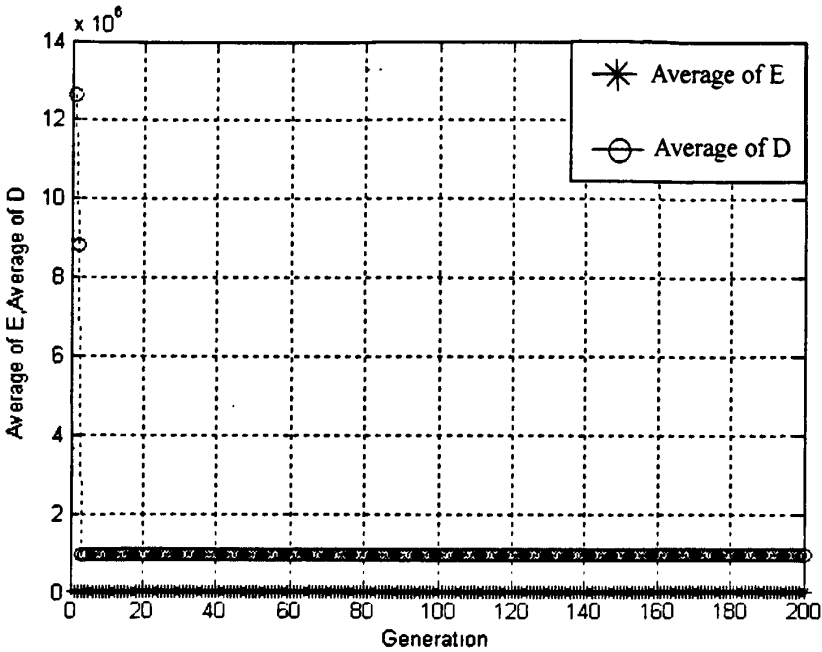


图 3-12 各代最优解的收敛过程

Fig. 3-12 The convergence process of the best solutions in each generation

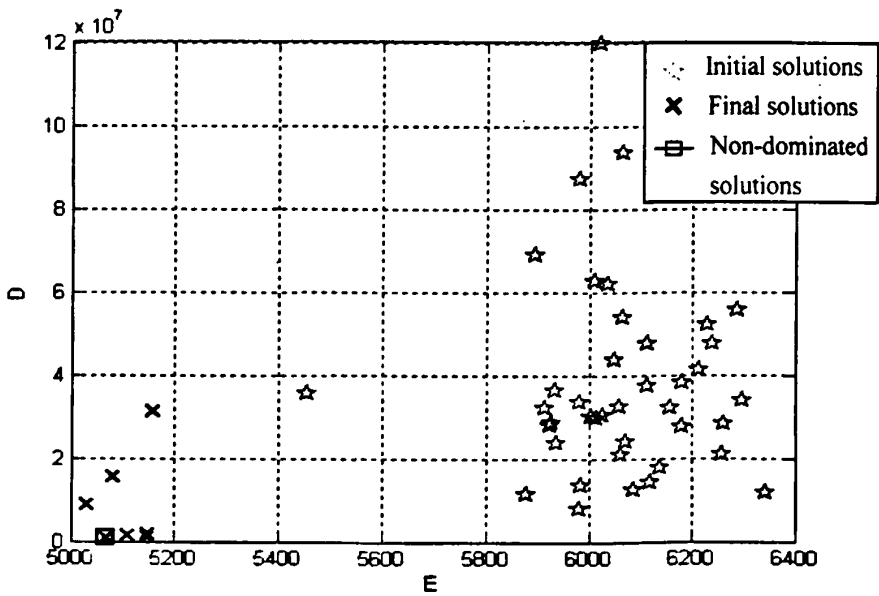


图 3-13 构建五个生产单元时各种解的分布图

Fig. 3-13 The distribution of various solutions for the instance with 5 cells

4 面向 Seru 生产的生产任务分配决策研究

4.1 生产任务分配问题的研究进展与本论文的注目点

传统单元化生产产生之前,市场需求低迷,为了能够利用固有的资源生产品种尽可能多的产品,大多数制造企业都采用具备高柔性的 job shop 形式组织生产资源。从 20 世纪 50 年代初期开始,大量文献从多个相关方面对 job shop 生产调度这一问题进行了系统详细的研究,这些研究从具体研究内容上看,主要是从订单投放、到期日分配、优先级分配和工人分配这四个方面展开的^[49]。生产任务分配问题作为生产计划与控制的重要决策之一,对单元化生产系统的正常运行和实施效果都有很大影响,然而该问题的研究并不像生产单元构建问题那样系统,研究成果也相对较少。

生产任务分配和生产任务排序通常放在一起考虑,先将生产任务分配给各生产单元,然后对各生产单元内的产品进行排序,确定每个产品的开始和结束时间。影响生产任务排序的因素包括生产单元构成、产品交付期、生产准备完毕时间和加工时间。常用的生产任务排序法则包括加工时间最短任务优先原则、加工时间最长任务优先原则、交付期最早任务优先原则等。对生产任务进行合理排序能够有效减少最大完工时间。对于没有考虑产品交付期的问题,生产任务排序通常将相似度较高的产品安排在一起连续生产,以减少人员移动和人员总数,从而降低劳动力成本。对于考虑产品交付期的问题,生产任务排序主要根据产品的换装时间和加工周期来确定产品的加工顺序,以减少延迟任务的数量,进而保证产品能够按时交货,提升客户满意度。但减少延迟任务和减少劳动力成本是两个相互矛盾的目标,若要减少延迟就需增加可用生产时间,增加工人数量。

从生产能力来源角度可以将生产单元分为机器密集型和劳动密集型两种。机器密集型生产单元的生产能力由机器数量决定,工人操作对其影响甚微,机器加工能力和时间的可变性较低,因此生产任务分配问题也比较简单。针对单元化生产方式的生产任务分配决策大多以劳动密集型生产单元为研究对象。劳动密集型生产单元大都由简易轻型设备组成,生产单元中工人数量通常大于操作技能总数,生产过程以工人为主导,设备起到辅助生产的作用。由于工人数量大于操作技能总数,每一项操作技能都由多个工人共同负责,因此,完成操作需要多个相同的机器和设备。劳动密集型生产单元常见于服饰、珠宝加工行业以及一些装配工厂。在处理劳动密集型生产单元的生产任务分配问题时,首先要确定每种产品的生产率,可用的方法包括瓶颈工序分析、生产线平衡、排队论方法、系统仿真。负责每项操作技能的工人数量由所生产的产品各工序的加工时间决定,这直接影响生产单元的产量。由于不同产品所需的工序不同,且各工序的加工时间也不同,所以在不同计划期内需求的变化会导致生产单元内的生产人数以及给各工序分配的工人数量的差异,这就产生了生产单元内和生产单元间的人员移动。在处理劳动密集型生产单元的生产任务分配问题时,通常要通过产品排序来尽可能减少人员移动距离。劳动密集型生产单元下的生产任务分配问题通常都没有考虑生产单元中可用技能与产品生产所需技能的匹配关系。

在以完成装配任务为主的生产单元中,生产能力是由生产单元内工人的数量、操作技能、熟练程度、加工时间等决定的。人员分配决策影响了生产单元生产能力的初值,生产任务分配决策是生产能力消耗决策,两种决策的结果相互影响,决定了生产单元的产量。为了能够实现生产能力的供需平衡,这两种决策通常都是同时考虑,有时也按照先后顺序考虑。在劳动密集型生产单元中,工人的表现与生产任务分配方案一样重要,都对生产单元性能有较大影响,所以工人分配与生产任务分配决策通常放在一起研究。

考虑到 Seru 生产与 job shop 机理的本质差异,与传统单元化生产的在实施和运作模式上的差别,本论文着眼于研究面向 Seru 生产的生产任务分配问题。本论文以 Seru 生产中最常见的装配生产单元为研究对象,装配生产单元主要负责将零部件组装为产成品。本论文考虑了产品交付期,为了保证按时交货,在分配生产任务的同时还给出了各生产单元内的生产顺序,明确了每类产品的生产开始时间和结束时间。

4.2 面向 Seru 生产的生产任务分配问题描述

本论文研究的面向 Seru 生产的生产任务分配问题主要解决如何将一个给定计划期内的生产任务分派到已经分配好工人的生产单元内,并确定每个生产单元的工作量和产品加工顺序。论文从四个维度定义生产单元和产品类,这四个维度分别是操作技能(部件种类)、技能熟练程度(品质等级)、装配时间(交付期)和加工成本(产品报价)。问题中涉及的主要决策包括生产任务分配、生产任务排序和订单接收。

生产任务是由销售部门提供的客户订单数据,客户订单中包含的所有产品根据包含的部件种类,对各部件的品质要求、交付期和产品报价这四个维度的属性划分为类,同一类产品四个维度的属性完全一致,不同类产品间至少有一个维度的属性存在差异。计划部门根据各类产品和生产单元的匹配程度确定哪类产品可以安排生产,哪类产品无法按照客户的要求完成生产。每个产品由多个部件组成,根据客户定制需求确定部件种类以及对各部件的品质要求。

生产单元是具备完结性的装配生产单元,这种生产单元以能够操作多项技能的多能工为核心,多能工在必要设备的辅助下完成产品的整个装配过程。生产单元四个维度的属性由生产单元内分配的多能工的能力决定,每个生产单元中包含多名多能工,每个多能工的操作技能组合及其熟练程度都不同,所以完成相应操作所需的时间和成本都不同。为了更好地体现工人操作熟练程度对生产任务分配决策的影响,问题中考虑的操作时间不是标准时间,而是根据工人操作熟练程度确定的基于技能水平的操作时间。工人的熟练程度越高,操作时间就越短,相应的支付给工人的操作成本就越高。每个多能工掌握多项技能,可能存在多个工人掌握同种技能,这种情况下,应选择技能熟练程度最高的工人来负责操作该项技能。员工掌握的技能与产品部件一一对应,具备生产某种部件的技能的工人能够完成装配该部件的一系列作业。同时,员工技能熟练程度决定了其装配的部件是否能够达到客户指定的品质要求等级。员工装配时间决定产品的生产周期,加工成本是给完成某种技能

的工人单位时间内支付的工资。

每个生产单元通常能够生产至少一种产品,对于每类产品来说,也可能存在多个生产单元能够胜任相应的生产任务。由于在给定计划期内生产单元的可用生产能力有限,所以本文研究的问题中允许进行批量分割,但仅当某类产品需求量太大,现有生产单元的可用生产能力不足时才能进行拆分,并将拆出部分分给其他生产单元。为了避免生产延迟,未被分配的各类产品按照交付期最早(EDD)原则排序,因此,交付期最早的一类产品会被安排在队列的最前端,能够尽早被分配。

在分配产品时,首先选择出工人操作技能和熟练程度能够满足产品部件种类和品质要求等级的生产单元,接着将这些生产单元按照生产成本排列为非减数列,最后根据产品交付期、生产周期和可用生产能力确定每类产品的生产任务分配给多少个生产单元以及各生产单元的工作量。Süer 和 Dagli^[66]将这种分配方法称为产品优先方法。当一类产品的批量过大,生产单元的生产能力不足时,为了提高生产单元的利用率,可以将该产品拆分为两个部分,把超过生产单元能力的部分视为一个新的批次重新放入等待队列队首,重新寻找合适的生产单元进行分配。这种批量分割规则借鉴了 Süer 等人^[66]提出的“适时拆分产品族”生产任务分配算法的思想。按照这种思路拆分产品批次能够更好地利用生产单元能力、降低停工时间并使工作量分配得相对均衡。当生产任务分配完毕后,并不是所有产品类都能找到合适的生产单元,如果客户愿意做出让步,这些调整后的产品类将在下一个计划期安排分配。图 4-1 描述了本论文研究的生产任务分配问题,从图中可以看出问题最终得到的结论包括哪类产品无法安排生产、各类产品分配到哪些生产单元及其工作量、以及生产单元内各类产品的加工顺序。

4.3 面向 Seru 生产的生产任务分配决策模型

本节建立了一个非线性数学规划模型来解决论文研究的生产任务分配问题,数学模型的目标函数是可变生产成本最小化,可变生产成本包括原材料成本、换装成本、加工成本和生产单元维护成本。

4.3.1 问题假设

- (1) 市场需求是确定型的。
- (2) 仅当生产单元可用生产能力不足时,一类产品可分割为两批并分配到不同的生产单元中生产。
- (3) 一类产品进行批量分割后,分割后的各部分必须全部分配给合适的生产单元且保证全部都在给定的交付期之前完成生产,一类产品不允许只生产一部分。
- (4) 生产单元具备完结性,能够从头到尾完成所分配产品的所有工序,在生产过程中不允许发生生产单元间物料移动。
- (5) 一个计划期内,每个多能工只能负责生产一种部件,并完成该部件的所有装配任务。

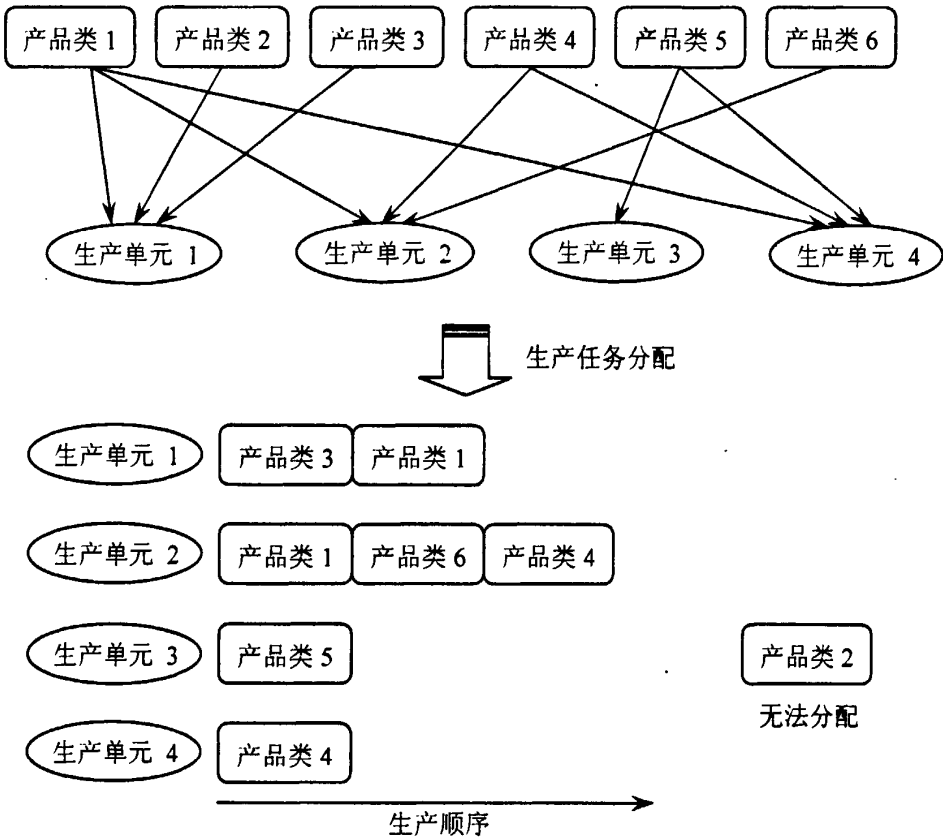


图 4-1 生产任务分配问题示意图

Fig. 4-1 The schematic diagram of cell loading problem

4.3.2 符号定义

- 输入参数:

i 生产单元编号, $i=1,2,\dots,C$;

j 产品类别编号, $j=1,2,\dots,P$;

k 产品部件种类数 (或生产单元技能种类数), $k=1,2,\dots,N$;

Q_j j 类产品的需求量;

s_i 生产单元 i 中工人的操作技能组合, $s_i=(s_i^1,s_i^2,\dots,s_i^k,\dots,s_i^N)$;

q_i 生产单元 i 中各项操作技能的熟练程度, $q_i=(q_i^1,q_i^2,\dots,q_i^k,\dots,q_i^N)$;

t_i 生产单元 i 中的工人在其所处的熟练程度下操作各项技能的单件装配时间

$$t_i=(t_i^1,t_i^2,\dots,t_i^k,\dots,t_i^N);$$

c_i 生产单元 i 中的工人在其所处的熟练程度下操作各项技能的单件装配成本,

$$c_i=(c_i^1,c_i^2,\dots,c_i^k,\dots,c_i^N);$$

c_i' 生产单元 i 内设备和工具的单位时间维护成本;

F_i 生产单元 i 的属性集合, $F_i=(s_i,q_i,t_i,c_i+c_i')$;

f_j j 类产品的部件组合, $f_j=(f_j^1,f_j^2,\dots,f_j^k,\dots,f_j^N)$;

e_j j 类产品中各部件的品质等级, $e_j=(e_j^1,e_j^2,\dots,e_j^k,\dots,e_j^N)$;

D_j j 类产品的交付期;

p_j j 类产品的报价;

O_j j 类产品的属性集合, $O_j = (f_j, e_j, D_j, p_j)$;

γ_{ij} j 类产品在生产单元 i 中生产时所需满足的利润率;

T 生产单元在一个计划期内的可用生产能力上限;

C_{ij} j 类产品在生产单元 i 中生产时所需的换装成本;

t_{ij} j 类产品在生产单元 i 中生产时所需的换装时间;

C'_j j 类产品的单件原材料成本。

• 辅助二进制变量:

$$s_i^k = \begin{cases} 1, & \text{当生产单元 } i \text{ 中工人具备技能 } k \text{ 时;} \\ 0, & \text{其他。} \end{cases}$$

$$f_j^k = \begin{cases} 1, & \text{当 } j \text{ 类产品包含部件 } k \text{ 时;} \\ 0, & \text{其他。} \end{cases}$$

$$g_{ij} = \begin{cases} 1, & \text{当 } j \text{ 类产品被分配到生产单元 } i \text{ 时;} \\ 0, & \text{其他。} \end{cases}$$

• 决策变量:

T_{ij} j 类产品在生产单元 i 中的换装启动时间;

Q_{ij} j 类产品分配到生产单元 i 中的产量。

• 输出参数:

Γ 生产任务分配方案。

4.3.3 数学模型

Minimize

$$C = \sum_{i=1}^C \sum_{j=1}^P \left\{ C'_j Q_{ij} g_{ij} + C_{ij} g_{ij} + Q_{ij} \cdot \sum_{k=1}^N c_i^k t_i^k f_j^k g_{ij} + c_i \left[t_{ij} g_{ij} + (Q_{ij} - 1) \cdot \max_{k=1}^N (t_i^k f_j^k g_{ij}) + \sum_{k=1}^N t_i^k f_j^k g_{ij} \right] \right\} \quad (4.1)$$

s. t.

$$s_i^k \geq f_j^k \quad \forall i, j, k, \text{ 且 } g_{ij} = 1 \quad (4.2)$$

$$q_i^k \geq e_j^k \quad \forall i, j, k, \text{ 且 } g_{ij} = 1 \quad (4.3)$$

$$C'_j + \sum_{k=1}^N c_i^k t_i^k f_j^k + \frac{1}{Q_{ij}} \cdot \left\{ C_{ij} + c_i \left[t_{ij} + (Q_{ij} - 1) \cdot \max_{k=1}^N (t_i^k f_j^k) + \sum_{k=1}^N t_i^k f_j^k \right] \right\} \leq \frac{p_j}{1 + \gamma_{ij}} \quad \forall i, j, \text{ 且 } g_{ij} = 1 \quad (4.4)$$

$$T_{ij} + t_{ij} + (Q_{ij} - 1) \cdot \max_{k=1}^N (t_i^k f_j^k) + \sum_{k=1}^N t_i^k f_j^k - D_j \leq 0 \quad \forall i, j, \text{ 且 } g_{ij} = 1 \quad (4.5)$$

$$T \geq \sum_{j=1}^P g_{ij} \left[t_{ij} + (Q_{ij} - 1) \cdot \max_{k=1}^N (t_i^k f_j^k) + \sum_{k=1}^N t_i^k f_j^k \right] \quad \forall i(4.6)$$

数学模型中的目标函数追求所有生产单元在一个计划期内可变生产成本最小化,可变生产成本由原材料成本、换装成本、加工成本和生产单元维护成本四个部分组成。由于产品是按照不同类别组织生产的,所以每当前一类产品结束生产后,都要为下一类投入的产品进行换装。加工成本是一个计划期内需要支付给工人的工资,它是根据所有生产单元内全部工人用于装配生产的时间和其单位时间工资确定的。生产单元维护成本除用于生产单元内操作台、工具、夹具等的维修和保养外,还用于将生产单元内多能工的技能熟练程度维持在一定范围内,此项成本与生产单元的运行时间成正比。

公式(4.2)和(4.3)确保负责装配某类产品的生产单元必须包含能够完成此类产品各部件装配任务所对应的全部技能,同时,技能水平还要能够满足各部件的品质要求等级。公式(4.4)表明,对于一类产品,若一个生产单元装配该类产品的可变生产成本和产品报价能够满足企业利润率要求,该类产品就能分配给此生产单元。公式(4.5)要求生产过程不允许出现延期,各类产品必须在交付期之前完成生产。公式(4.6)确保分配到每个生产单元的工作量不能超过生产单元在一个计划期内的最大可用生产能力。

4.4 基于按需分解订单法则的启发式算法开发

4.4.1 算法思想

本节给出一个启发式算法来求解生产任务分配问题的数学模型,算法基本思路为:首先根据产品的部件组成和品质要求和生产单元具备的加工技能和熟练水平确定每类产品的可用生产单元,考虑到本文研究的问题追求可变生产成本最小,故将这些可用生产单元按照生产一单位各类产品的可变生产成本排列为非减数列;各类产品将优先分配到可变生产成本较低的生产单元,每类产品分配到各生产单元的任务量由生产单元的可用生产能力决定,若生产单元可用生产能力不足,则进行批量分割;分配的任务量确定后,检查在某一生产单元内生产指定数量产品的时间和成本是否满足产品的交付期和企业的利润率,若满足,则将该类产品分配给这一生产单元,否则,寻找其他可用生产单元。

4.4.2 算法步骤

本论文开发的启发式算法具体步骤如下:

注:算法中 T_i^h , s , b 和 m 只是变量符号,没有任何实际意义,并不代表数学模型中定义的符号的含义。

步骤 1: 输入参数,变量初始化,设 $m=1$, $T_i^h=0$,其中 m 为已分配产品类的计数器, T_i^h 是生产单元 i 内生产的各类产品的完工时间, $i=1,2,\dots,C$, $h=1,2,\dots,P$;

步骤 2: 选出待分配队列中交付期最早的一类产品,设该类产品的类别编号为 g ,

$g \in \{1, 2, \dots, P\}$;

步骤 3: 根据 g 类产品部件组合中包含的部件种类及相应的品质要求等级, 找出所有满足公式(4.2)和(4.3)的生产单元, 并将这些生产单元的编号放入集合 R 中;

- 若 $R = \phi$, 输出“由于 Seru 生产系统不具备装配 g 类产品的能力, 故拒绝接收 g 类产品”, 并执行步骤 8;

- 若 $R \neq \phi$, 执行步骤 4;

步骤 4: 设集合 R 中共包含 n 个元素, 将集合 R 中各元素所对应的生产单元按照生产一单位 g 类产品所需的可变成本非减原则进行排序, 生产单元 i 生产一单位 g 类产品的可变成本为 $C_g' + \frac{C_g}{Q_g} + \sum_{k=1}^N c_i^k t_i^k f_g^k + c_i' \left[t_{ig} + \sum_{k=1}^N t_i^k f_g^k \right]$ 。给排序后集合 R 中的生产单元分配新的代

码, 从第一个生产单元开始, 各生产单元原编号对应的新代码分别为 r_1, r_2, \dots, r_n ;

步骤 5: 设 $b = 1$;

步骤 6: 记录 g 类产品在生产单元 r_b 中的换装启动时间, g 类产品的换装启动时间为已分配到生产单元 r_b 中的各类产品完工时间的最大值, $T_{r_b g} = \max \{T_{r_b}^1, T_{r_b}^2, \dots, T_{r_b}^m\}$;

步骤 7: 设 $Q_{r_b g}$ 是 Z 所能取得的最大整数, Z 必须满足下面两个公式, 即

$$T_{r_b g} + t_{r_b g} + (Z-1) \cdot \max_{k=1}^N (t_{r_b}^k f_g^k) + \sum_{k=1}^N t_{r_b}^k f_g^k - D_g \leq 0, \quad T \geq t_{r_b g} + (Z-1) \cdot \max_{k=1}^N (t_{r_b}^k f_g^k) + \sum_{k=1}^N t_{r_b}^k f_g^k;$$

- 若 $\sum_{s=1}^b Q_{r_s g} > Q_g$, 则 $Q_{r_b g} = Q_g - \sum_{s=1}^{b-1} Q_{r_s g}$;

- 若

$$\frac{1}{Q_{r_b g}} \cdot \left\{ C_{r_b g} + c_{r_b} \left[t_{r_b g} + (Q_{r_b g} - 1) \cdot \max_{k=1}^N (t_{r_b}^k f_g^k) + \sum_{k=1}^N t_{r_b}^k f_g^k \right] \right\} + C_g' + \sum_{k=1}^N c_{r_b}^k t_{r_b}^k f_g^k \leq \frac{P_g}{1 + \gamma_{r_b g}}$$

$$\text{则 } T_{r_b}^m = T_{r_b g} + (Q_{r_b g} - 1) \cdot \max_{k=1}^N (t_{r_b}^k f_g^k) + t_{r_b g} + \sum_{k=1}^N t_{r_b}^k f_g^k,$$

$$T = T - \left\{ t_{r_b g} + (Q_{r_b g} - 1) \cdot \max_{k=1}^N (t_{r_b}^k f_g^k) + \sum_{k=1}^N t_{r_b}^k f_g^k \right\}, \text{ 记录 } Q_{r_b g}, T_{r_b}^m \text{ 和 } T \text{ 的值, 接}$$

着执行步骤 8;

- 若

$$\frac{1}{Q_{r_b g}} \cdot \left\{ C_{r_b g} + c_{r_b} \left[t_{r_b g} + (Q_{r_b g} - 1) \cdot \max_{k=1}^N (t_{r_b}^k f_g^k) + \sum_{k=1}^N t_{r_b}^k f_g^k \right] \right\} + C_g' + \sum_{k=1}^N c_{r_b}^k t_{r_b}^k f_g^k > \frac{P_g}{1 + \gamma_{r_b g}}$$

则 $b = b + 1$;

- 若 $b \leq n$, 执行步骤 6;

- 若 $b > n$, 输出“由于 Seru 生产系统不具备装配 g 类产品的能力, 故拒绝接收 g 类产品”, 并执行步骤 8;

- 若 $\sum_{s=1}^b Q_{r,s} \leq Q_g$;

- 若

$$\frac{1}{Q_{r,g}} \cdot \left\{ C_{r,g} + c_{r,g} \left[t_{r,g} + (Q_{r,g} - 1) \cdot \max_{k=1}^N (t_{r,g}^k f_g^k) + \sum_{k=1}^N t_{r,g}^k f_g^k \right] \right\} + C_g + \sum_{k=1}^N c_{r,g}^k t_{r,g}^k f_g^k \leq \frac{P_g}{1 + \gamma_{r,g}}$$

$$\text{则 } T_{r,g}^m = T_{r,g} + (Q_{r,g} - 1) \cdot \max_{k=1}^N (t_{r,g}^k f_g^k) + t_{r,g} + \sum_{k=1}^N t_{r,g}^k f_g^k,$$

$$T = T - \left\{ t_{r,g} + (Q_{r,g} - 1) \cdot \max_{k=1}^N (t_{r,g}^k f_g^k) + \sum_{k=1}^N t_{r,g}^k f_g^k \right\}, \text{ 记录 } Q_{r,g}, T_{r,g}^m \text{ 和 } T \text{ 的值, 接着,}$$

$$b = b + 1;$$

- 若 $b \leq n$, 执行步骤 6;
- 若 $b > n$, 输出“由于 Seru 生产系统不具备装配 g 类产品的能力, 故拒绝接收 g 类产品”, 并执行步骤 8;

- 若

$$\frac{1}{Q_{r,g}} \cdot \left\{ C_{r,g} + c_{r,g} \left[t_{r,g} + (Q_{r,g} - 1) \cdot \max_{k=1}^N (t_{r,g}^k f_g^k) + \sum_{k=1}^N t_{r,g}^k f_g^k \right] \right\} + C_g + \sum_{k=1}^N c_{r,g}^k t_{r,g}^k f_g^k > \frac{P_g}{1 + \gamma_{r,g}}$$

$$\text{则 } b = b + 1;$$

- 若 $b \leq n$, 执行步骤 6;
- 若 $b > n$, 输出“由于 Seru 生产系统不具备装配 g 类产品的能力, 故拒绝接收 g 类产品”, 并执行步骤 8;

步骤 8: $m = m + 1$;

- 若 $m \leq P$, 执行步骤 2;
- 若 $m > P$, 执行步骤 9;

步骤 9: 根据步骤 6 和步骤 7 中确定的生产任务分配方案输出 T_{ij} 和 Q_{ij} , 其中 $i \in \{1, 2, \dots, C\}$, $j \in \{1, 2, \dots, P\}$ 。

虽然本论文中允许在生产单元生产能力不足时, 对一类产品进行批量分割, 但是, 分割后的各部分必须都分配给合适的生产单元。若生产任务分配方案中, 某类产品有一部分分配了, 剩余部分没有分配; 则该方案是无效的。因为对某类产品进行批量分割时是根据生产单元的可用生产能力确定分割后各部分的批量的, 若只生产一部分, 将可能导致生产出的产品不符合客户订单中的需求量, 既增加了库存成本, 又无法保证准时发货。若出现这种情况, 则认为被部分分配的这类产品无法在当期生产, 故将其从生产任务中抽出, 对剩余的生产任务重新进行分配, 以找到最佳分配方案。对于那些无法安排在当期生产的产品来说, 如果客户愿意对产品的部件组合、品质要求、价格和交付期等进行调整, 那么调整后的订单将会在下个计划期安排生产。

4.5 算例分析

4.5.1 算例描述

为了验证本论文为解决面向 Seru 生产的生产任务分配问题而建立的数学模型的正确性以及用于模型求解的启发式算法的有效性，本节利用随机产生的数据来模拟一个静态的、需求确定的生产任务分配算例。算例分析过程为：首先使用 Visual Basic 程序语言将上节给出的启发式算法编码，并在 Visual Basic 中实现；然后，根据算例特征，适当调整 Sürer 等人^[65]提出的“适时拆分子产品族”生产任务分配算法，并用调整后的算法求解本算例；最后，对比两种算法的计算结果。

本节所给算例的背景是一个包含四个装配生产单元的小型 Seru 生产系统。每个生产单元能够装配相应产品类包含的所有部件，整个 Seru 生产系统中各生产单元能够操作的技能一共有八种，一种技能能够装配一种指定的部件，相应的，每类产品是由八种部件中的部分或全部任意组合构成的。一个计划期为一周，按一周五天，每天工作 8 小时计算，计划期内每个生产单元的可用生产能力上限为 2400min，生产系统初始时间设为 2011-3-7。表 4-1 给出了每类产品各属性的信息，表 4-2 给出了各生产单元的初始状态，表 4-3 给出了与产品类和生产单元同时相关的数据。

表 4-1 各类产品属性表
Table 4-1 The list of attributes of product types

产品类	属性	部件 1	部件 2	部件 3	部件 4	部件 5	部件 6	部件 7	部件 8
1	部件集	1	0	1	0	1	1	0	1
	品质等级	0.9	0	1.0	0	0.6	0.9	0	0.7
	交付期	2011-3-14							
	单价（元/件）	532							
	原材料成本（元/件）	73							
	批量（件）	250							
2	部件集	0	1	1	0	0	1	0	1
	品质等级	0	0.8	0.6	0	0	0.4	0	0.9
	交付期	2011-3-12							
	单价（元/件）	437							
	原材料成本（元/件）	64							
	批量（件）	390							
3	部件集	1	0	0	1	1	1	0	1
	品质等级	0.5	0	0	0.8	0.4	0.8	0	0.4
	交付期	2011-3-10							
	单价（元/件）	582							
	原材料成本（元/件）	97							
	批量（件）	140							

续表 4-1

4	部件集	1	0	1	1	0	1	1	1
	品质等级	0.8	0	0.6	0.7	0	0.9	0.3	0.7
	交付期	2011-3-9							
	单价（元/件）	689							
	原材料成本（元/件）	128							
	批量（件）	240							
5	部件集	1	1	1	0	1	0	0	1
	品质等级	0.6	0.4	0.8	0	0.9	0	0	0.5
	交付期	2011-3-12							
	单价（元/件）	564							
	原材料成本（元/件）	102							
	批量（件）	170							
6	部件集	1	1	1	0	1	1	0	0
	品质等级	0.7	0.5	1.0	0	1.0	0.7	0	0
	交付期	2011-3-13							
	单价（元/件）	573							
	原材料成本（元/件）	75							
	批量（件）	230							
7	部件集	1	1	0	1	0	0	1	1
	品质等级	1.0	0.6	0	0.8	0	0	0.6	0.7
	交付期	2011-3-11							
	单价（元/件）	607							
	原材料成本（元/件）	116							
	批量（件）	160							
8	部件集	0	1	1	1	1	0	1	0
	品质等级	0	1.0	0.3	0.8	0.6	0	0.5	0
	交付期	2011-3-8							
	单价（元/件）	719							
	原材料成本（元/件）	204							
	批量（件）	110							

表 4-2 生产单元初始状态表

Table 4-2 The list of attributes of cells

生产单元	属性	技能 1	技能 2	技能 3	技能 4	技能 5	技能 6	技能 7	技能 8
生产单元 1	技能集	1	1	1	0	1	1	0	1
	熟练程度	1.0	0.7	1.0	0	1.0	0.9	0	0.8
	单位装配时间（分/件）	4.5	2.8	3.6	0	1.6	2.9	0	3.4
	单位装配成本（元/分）	12.2	19.8	18.0	0	16.7	16.5	0	20.0
	维护成本（元/分）	13.6							
生产单元 2	技能集	0	1	1	1	1	1	1	1
	熟练程度	0	1.0	0.7	0.9	0.7	0.6	0.5	0.9
	单位装配时间（分/件）	0	2.6	3.2	3.3	2.4	2.1	4.0	3.5
	单位装配成本（元/分）	0	19.8	18.0	18.2	20.0	13.8	17.3	12.8
	维护成本（元/分）	10.5							

续表 4-2

生产单元 3	技能集	1	1	1	1	0	1	1	1
	熟练程度	1.0	0.9	0.6	0.9	0	1.0	0.7	1.0
	单位装配时间(分/件)	4.2	2.8	2.9	3.7	0	2.6	3.4	2.6
	单位装配成本(元/分)	13.5	17.6	18	15.7	0	14.3	12.9	18.7
	维护成本(元/分)	14.2							
生产单元 4	技能集	1	1	1	1	1	1	1	1
	熟练程度	0.9	1.0	1.0	0.8	0.9	0.9	0.7	0.8
	单位装配时间(分/件)	5.0	4.6	2.4	3.5	2.7	1.8	3.0	2.5
	单位装配成本(元/分)	13.5	20.0	18.4	19.2	16.8	9.4	16.3	17.8
	维护成本(元/分)	15.0							

表 4-3 产品换装及利润率表

Table 4-3 The list of setup data and profit rate

产品类	属性	生产单元 1	生产单元 2	生产单元 3	生产单元 4
1	利润率	0.23	0.36	0.19	0.25
	换装成本(元/次)	66	48	75.5	70
	换装时间(分)	12.5	13.0	11.8	12.2
2	利润率	0.21	0.33	0.18	0.27
	换装成本(元/次)	46.5	58	76	71.5
	换装时间(分)	13.8	12.6	14.2	12.3
3	利润率	0.22	0.31	0.21	0.30
	换装成本(元/次)	71	82	54.5	67.5
	换装时间(分)	11.6	12.4	13.0	13.6
4	利润率	0.20	0.29	0.28	0.22
	换装成本(元/次)	68.5	57.5	81	92
	换装时间(分)	12.8	14.1	11.6	13.5
5	利润率	0.19	0.26	0.20	0.26
	换装成本(元/次)	80	76	70.5	82.5
	换装时间(分)	11.9	15.6	16.2	14.3
6	利润率	0.25	0.34	0.23	0.34
	换装成本(元/次)	62	43	82.5	66
	换装时间(分)	12.5	13.4	16.3	16.7
7	利润率	0.22	0.29	0.26	0.28
	换装成本(元/次)	59.5	84	86	52.5
	换装时间(分)	13.8	12.7	14.2	12.2
8	利润率	0.21	0.25	0.30	0.28
	换装成本(元/次)	78	91.5	58.5	67
	换装时间(分)	14.3	11.8	15.9	13.7

4.5.2 算例结果对比分析

利用本论文给出的算法计算算例，首先将各类产品按照交付期最早(EDD)原则进行排序，按照交付期由近到远排序后得到的产品类序列为：8类产品、4类产品、3类产品、7类产品、2类产品、5类产品、6类产品、1类产品。接着，根据各类产品所包含的部件

种类和品质等级要求,为每类产品找出满足公式(4.2)和(4.3)的可用生产单元,并将这些生产单元按照生产一单位指定类产品所需的可变成本非减原则进行排序,结果如表 4-4 所示。根据上节给出的算法步骤进行分配,本算例的生产任务分配方案和工作量分布情况分别如表 4-5 和图 4-2 所示。

表 4-4 各类产品与生产单元的匹配关系表

Table 4-4 The matching relationship between product types and cells

产品类	可用生产单元
1 类产品	生产单元 4, 生产单元 1
2 类产品	生产单元 2, 生产单元 3
3 类产品	生产单元 4
4 类产品	生产单元 3, 生产单元 4
5 类产品	生产单元 1, 生产单元 4
6 类产品	生产单元 1
7 类产品	生产单元 3
8 类产品	生产单元 2, 生产单元 4

表 4-5 本文算法得到的生产任务分配方案

Table 4-5 The cell loading plan given by the proposed algorithm

产品类	生产单元编号	批量 (件)	换装启动时间
1	4	250	2011-3-8 14:04
2	2	390	2011-3-7 17:44
3	4	140	2011-3-7 8:00
4	3	240	2011-3-7 8:00
5	1	170	2011-3-7 8:00
6	1	230	2011-3-8 15:09
7	3	160	2011-3-9 9:15
8	2	110	2011-3-7 8:00

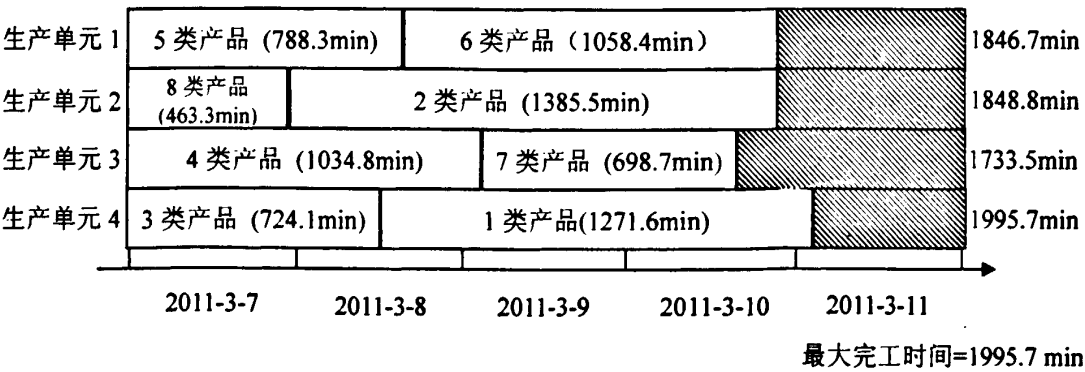


图 4-2 本文算法得到的工作量分布图

Fig. 4-2 The workload distribution of cell loading plan given by the proposed algorithm

本论文提出的用于解决面向 Seru 生产的生产任务分配问题的算法在处理批量分割问题时的基本思路与 Süer 等人^[65]提出的“适时拆分子产品族”生产任务分配算法的思想相

近。只不过 Süer 等人提出的算法是针对劳动密集型生产单元下的生产任务分配问题的，且在他的问题中首先将生产任务划分为产品族，然后以产品族为单位分配生产任务。本文研究的问题中虽没有涉及产品族，但是产品订单中的不同产品根据四个维度的属性值划分成了不同类。为了便于对两种算法解决同一个算例的结果进行对比，现根据本文所给算例的特征和考虑的因素，对 Süer 等人提出的算法进行调整，调整后的具体算法步骤为：

步骤 1：对每类产品，找出所有能够生产该类产品的生产单元，并计算每类产品在每个可用生产单元内的总流程时间，即换装时间与加工周期之和；

步骤 2：根据步骤 1 中计算的总流程时间，将每类产品的可用生产单元按照总流程时间非减原则排序，即每类产品将优先分配给总流程时间较短的可用生产单元中；

步骤 3：根据每类产品的第一个可用生产单元，确定各生产单元能够生产的产品类列表，建立第一分配方案，并将每个生产单元能够生产的各类产品按照加工时间最长（LPT）原则进行排序，排序后总流程时间最长的产品类位于队列前端，将被优先分配；根据每类产品的第二个可用生产单元再次确定各生产单元能够生产的产品类列表，建立第二分配方案，并对各类产品进行排序；以此类推，根据各类产品可用生产单元列表中生产单元的数量建立相应数量的分配方案；

步骤 4：从第一分配方案开始，选择一个没被处理过的分配方案，若所有方案都被处理过，则执行步骤 10；

步骤 5：从选择的分配方案中选择一个未处理的生产单元 i ，执行步骤 6；若所有生产单元都被处理，执行步骤 4；

步骤 6：从生产单元 i 对应的产品类集合中选择一个产品类 j ，执行步骤 7；

步骤 7：若 j 类产品的全部生产任务已经分配给其他生产单元，则执行步骤 6；否则，执行步骤 8；

步骤 8：在 j 类产品的单价下，如果 j 类产品在生产单元 i 中生产的可变生产成本能够满足相应的利润要求，则执行步骤 9；否则，执行步骤 6；

步骤 9：若 j 类产品的总流程时间不超过生产单元 i 的可用生产能力，则将 j 类产品全部分配给生产单元 i ，接着执行步骤 6；否则，将 j 类产品拆分为两批 $G1$ 和 $G2$ ， $G1$ 的批量根据生产单元 i 的可用生产能力确定，将 $G1$ 分配给生产单元 i ，将 j 类产品标记为未完全分配并记录剩余工作量，然后执行步骤 6；

步骤 10：拒绝接收那些无法分配的产品类。

根据以上生产任务分配算法对本论文给出的算例进行求解，表 4-6 给出了各类产品的可用生产单元列表，表 4-7 给出了各生产单元的各级分配方案，最终得到的生产任务分配结果如图 4-3 和表 4-8 所示。由于生产单元 1 的剩余生产能力不足，需将 5 类产品拆分成两批，但是由于 5 类产品在生产单元 4 中生产的成本不能满足给定的利润率要求，故 5 类产品无法分配。7 类产品只有一个可用生产单元，但由于生产单元 3 的可用生产能力不足，故 7 类产品也无法分配。

表 4-6 各类产品与生产单元的匹配关系表

Table 4-6 The matching relationship between product types and cells

产品类	可用生产单元
1 类产品	生产单元 1, 生产单元 4
2 类产品	生产单元 3, 生产单元 2
3 类产品	生产单元 4
4 类产品	生产单元 3, 生产单元 4
5 类产品	生产单元 1, 生产单元 4
6 类产品	生产单元 1
7 类产品	生产单元 3
8 类产品	生产单元 2, 生产单元 4

表 4-7 产品类分配预案

Table 4-7 The initial cell loading plan of product types

生产单元编号	第一分配方案	第二分配方案
1	产品类 1, 产品类 6, 产品类 5	
2	产品类 8	产品类 2
3	产品类 2, 产品类 4, 产品类 7	
4	产品类 3	产品类 1, 产品类 4, 产品类 5, 产品类 8

表 4-8 Sürer 等人算法得到的生产任务分配方案

Table 4-8 The cell loading plan given by Sürer's algorithm

产品类	生产单元编号	批量 (件)	换装启动时间
1	1	250	2011-3-7 8:00
2	3	390	2011-3-7 8:00
3	4	140	2011-3-7 8:00
4	3	240	2011-3-9 11:14
5	无法分配		
6	1	230	2011-3-9 11:10
7	无法分配		
8	2	110	2011-3-7 8:00

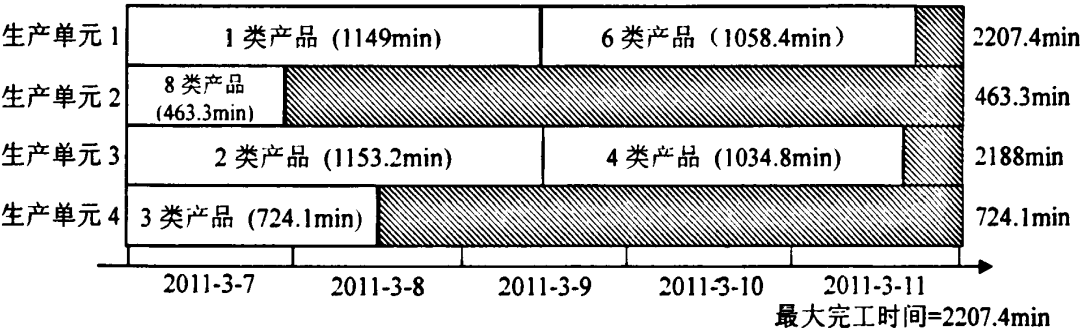


图 4-3 Sürer 等人算法得到的生产任务分布图

Fig. 4-3 The workload distribution of cell loading plan given by Sürer's algorithm

根据上文中两种算法对同一算例给出的生产任务分配方案可得,本文算法所得方案将全部产品都分配到合适的生产单元中,可变生产成本为 669558.22 元,最大完工时间为 1995.7min,而 Süer 等人的算法所得方案未能将全部产品都分配到合适的生产单元中,可变生产成本为 537000.99 元,最大完工时间为 2207.4min。虽然 Süer 等人的算法所得方案的可变生产成本较低,但该方案在生产单元仍有大量生产能力没有使用的情况下,存在两类产品无法得到合理分配,这些未能分配的产品不但会影响企业生产和盈利,还会降低生产系统柔性。同时, Süer 等人的算法所得方案虽然生产的产品类较少,可是整个生产过程的最大完工时间却明显大于本论文所给算法得出的方案。

由于本文给出算法在分配产品时将所有产品类按照交付期由近到远的顺序排列,优先分配交付期早的产品类,这样就能够让交付期早的产品提早生产,避免生产延迟,保证各类产品按时交货。Süer 等人的算法将产品类按照总流程时间由长到短的顺序排列,他们认为这样做有助于减少最大完工时间、平衡工作量和提高生产单元能力利用率,同时能够保证加工周期较长的生产任务提前分配,保证了一类产品生产流程的完整性,避免了对大批量任务的多次拆分。这种方法在所有产品的交付期一致时是比较可行的,但对于本文给出的具备不同交付期的生产任务分配问题,这种算法得到的结果中可能会出现某些产品即使能够安排生产,却无法按时交货的现象。上节得到的分配方案中 4 类产品虽然分配给生产单元 3,但是该类产品在规定的交货最后日期才能开始投入生产。

两种算法所得生产任务分配方案中各生产单元的停工时间分布情况和生产单元能力利用率如图 4-4 和 4-5 所示。从图中可以看出,由于本文给出的算法将所有生产任务都分配给了合适的生产单元,故整个计划期内各生产单元的停工时间明显小于 Süer 等人算法得到的结果。相应的,生产单元能力利用率也较高。停工时间过长会严重影响工人的工作积极性。从图 4-5 可以看出,本文算法给出的方案中各个生产单元的工作量分配相对均衡,波动较小,而 Süer 等人算法给出的方案中生产单元 1,3 分配的工作量较多,而生产单元 2,4 分配的工作量较少,在整个计划期内大部分时间处于空闲状态。然而,5 类产品和 7 类产品却无法在这些空闲的生产单元内生产。综上所述,对于本论文研究的生产任务分配问题,本论文给出的算法比 Süer 等人的算法得到的分配方案更令人满意。

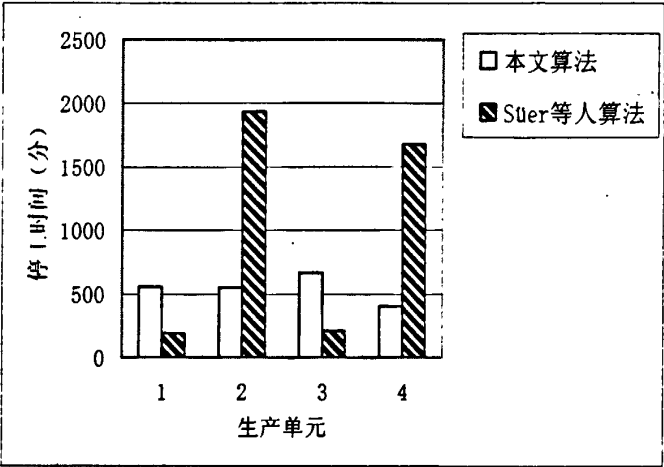


图 4-4 两种算法所得分配方案的停工时间对比图

Fig. 4-4 The comparison between idle time of cell loading plans given by the two algorithms

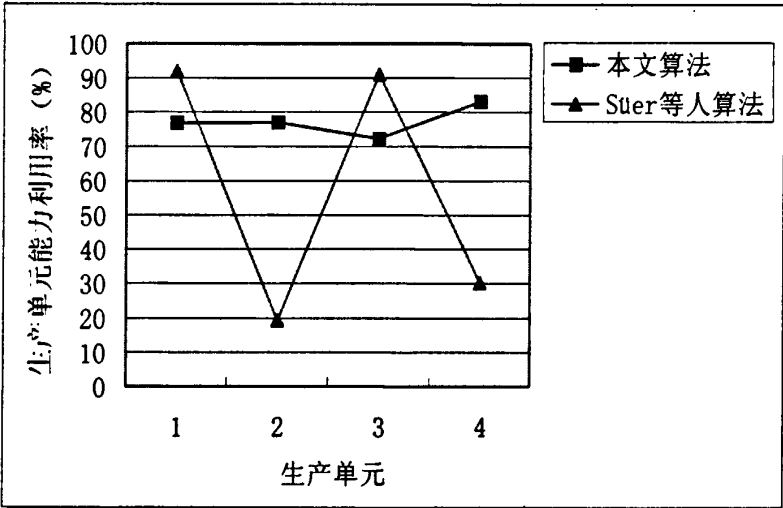


图 4-5 两种算法所得分配方案的生产单元能力利用率对比图

Fig. 4-5 The comparison between machine utilization of cell loading plans given by the two algorithms

5 结论与展望

5.1 结论

Seru 生产是 20 世纪 90 年代产生于日本索尼株式会社的一种先进生产方式, 这种生产方式兼具效率与柔性, 能够有效地应对产品需求多品种、小批量化带来的变化和影响。Seru 生产自产生以来, 在日本许多行业得到了广泛的应用并取得了良好的收益, 大量介绍 Seru 生产在日本企业实施经验的文章和专著相继出现。但是由于 Seru 生产产生的历史较短, 且与传统单元化生产方式存在许多相似之处, 导致 Seru 生产领域的研究较为滞后。现有学术文献中, 大量关于 Seru 生产的研究仅局限于对 Seru 生产实施经验的介绍和基础理论的阐述, 大量关键决策问题都尚未解决, Seru 生产的理论与方法体系远不完善。本论文在对大量相关文献进行分析的基础上, 重点研究了面向 Seru 生产的生产单元构建和生产任务分配两大关键决策。

生产单元构建决策是企业实施 Seru 生产要解决的首要决策, 它是 Seru 生产其他关键决策研究的基础, 对生产系统的整体设计和正常运行具有重要影响。本论文在充分把握生产单元构建问题国内外研究现状的基础上, 结合企业实践经验, 构建了符合 Seru 生产特征的生产单元构建决策模型。该模型从实现生产单元间工作量均衡的角度来研究生产单元构建问题, 综合分析同质机器和可替换工艺路线对生产单元构建问题的影响, 并考虑了加工时间、换装时间、加工顺序、产品需求量、机器加工能力、生产单元大小和生产单元数量等多个实际生产要素。考虑到多目标优化问题的求解难度, 本论文根据所建数学模型的特征和遗传算法相关理论, 开发了一个基于非支配排序遗传算法-II 的混合遗传算法。通过使用该算法对算例求解证实, 该算法能够在合理的时间内求得令人满意的解, 且计算结果充分体现了 Seru 生产的生产完结性特征。

生产任务分配问题是 Seru 生产下生产计划与控制领域的重要决策, 它对生产系统的时间性能指标具有重要影响。在先前研究中, 分配生产任务时通常只考虑产品工序和生产单元具备的技能之间的匹配。本论文在充分把握生产任务分配问题国内外研究现状的基础上, 以装配生产单元为研究对象, 构建了符合 Seru 生产特征的生产任务分配决策模型。为使生产任务得到更合理的分配, 生产能力得到充分利用, 模型中从四个维度定义生产单元和产品, 这四个维度分别是操作技能(部件种类)、技能熟练程度(品质等级)、装配时间(交付期)和加工成本(产品报价)。结合模型的特征, 本论文选取合适的批量分割规则, 开发了一个启发式算法对模型进行求解。通过对比本文提出算法和 Süer 等人提出算法对同一算例求解的计算结果可以看出, 本文给出的算法在最大完工时间、生产任务均衡性、生产能力利用率等方面表现较好。

5.2 创新点

本论文的创新点主要包括以下两点:

(1) 通过利用同质机器和可替换工艺路线, 彻底消除了生产单元间物料移动, 保证了生产单元的独立性和生产完结性。

(2) 首次从多维度属性来定义生产单元和产品, 通过实现生产单元和产品属性之间的紧密耦合, 提高了生产任务分配方案的准确性, 有利于充分利用生产能力。

5.3 展望

本论文虽然提出了解决面向 Seru 生产的生产单元构建和生产任务分配问题的方法, 但是这两个关键决策的理论与方法体系还远不完善, 仍有许多问题值得进一步深入研究。

(1) 本论文中研究的生产单元构建问题虽然彻底消除了生产单元间物料移动, 但是并没有考虑生产单元内移动对生产过程的影响。生产单元内部布局设计是生产单元构建方案的基础上对生产单元内包含的设备进行布局设计, 它直接影响着生产单元内物料移动的次数。因此, 在解决生产单元构建问题时考虑生产单元内部布局设计, 就能够同时消除生产单元间和生产单元内物料移动, 使生产单元构建方案更加完善。

(2) 本论文给出的生产任务分配方案只能明确分配到各生产单元的产品类和任务量, 并未考虑生产单元内工人的生产任务分配情况。Seru 生产是以人为中心的生产方式, 多能工是 Seru 生产实施的基础和前提。多能工是生产单元的提供者, 生产任务是生产能力的消耗者, 因此, 应当将生产任务分配决策和人员分配决策放在一起同时考虑, 这样才能更好地实现生产能力的供需平衡。

(3) 本论文研究的生产单元构建问题和生产任务分配问题中, 市场需求都是确定的、已知的。但是 Seru 生产所面对的是产品种类和需求量多变的多品种、小批量需求。为使研究内容更符合生产实际, 今后的研究中应该更加重视动态需求环境下相关决策的解决方法。

致谢

转眼之间，硕士研究生生涯即将结束。回忆起两年多的学习生活，既有甜，也有苦。在学习的过程中，尽管遇到了许多困难，碰到许多棘手的难题，但是在老师的指导和同学的帮助下，我通过自己的不断努力，最终克服了重重困难，为这段学习生活划上了圆满的句号。在此，我由衷地感谢帮助过我的老师和同学。

首先，我要感谢我的导师刘晨光教授这三年来对我学习和生活上给予的帮助与支持。在刘老师的悉心指导和谆谆教诲下，我很快地适应了研究生阶段的学习，并掌握了科学的文献查阅和论文撰写方法。在我遇到问题时，刘老师总会倾囊相授，认真为我答疑解惑。刘老师是我学习的榜样，三年来他刻苦钻研、追求创新的科研精神和兢兢业业的工作态度深深地影响了我。在此我谨向刘老师表达我最诚挚的谢意！

其次，我要感谢经济与管理学院的各位老师，感谢您们传授给我专业的科学知识和有效的学习方法，这为我快速适应学术研究工作打下了坚实的基础。

同时，我要感谢研究团队其他成员对我的帮助和鼓励，感谢同学们对我生活上无微不至的照顾，感谢家人朋友对我的关怀、理解和支持。

最后，我要衷心感谢诸位参与论文评审的老师。

参考文献

- 【1】Toffler, A.. Future Shock[M]. New York: Random House, 1970.
- 【2】Tsuru, T.. The Evolution and Refinement of Production Systems: the Diffusion of Seru Seisan in Japanese Enterprises (in Japanese) [M]. Tokyo: Nihon Hyoron Shya. 2001.
- 【3】Wemmerlöv, U., Hyer, N.L.. Cellular manufacturing in the US industry: a survey of users[J]. International Journal of Production Research, 1989, 27(9): 1511-1530.
- 【4】Wemmerlöv, U., Johnson, D.J.. Cellular manufacturing at 46 user plants: Implementation experiences and performance improvements[J]. International Journal of Production Research, 1997, 35: 29-49.
- 【5】Economic Research Institute. Cell and Innovation of Production System (in Japanese)[M]. Japan: Economic Research Institute, 1997.
- 【6】Iwamuro, H.. Cell Manufacturing System (in Japanese)[M]. Tokyo: Nikkan Kogyo Shimbun, 2002.
- 【7】Noguchi, H.. Production Involution in Japan (in Japanese)[M]. Tokyo: Nikkan Kogyo Shimbun, 2003.
- 【8】D&M Nikkei Mechanical. The Challenge of Canon-Part 3[J]. D&M Nikkei Mechanical, 2003, 588(September): 70-73.
- 【9】Kunii, I.. He put the flash back in Canon, how Fujio Mitarai's east-west style revived the company[J]. Bloomberg Business Week, 2002, Sep. 16: 40-42.
- 【10】Shinobu, C.. Post-Lean Production Systems: Toward an Adaptable Enterprise in the Age of Uncertainty (in Japanese)[M]. Tokyo: Bunnshin-Do, 2003.
- 【11】Liu, C.G., Lian, J., Yin, Y., Li, W.J.. Seru seisan — an innovation of the production management mode in Japan[J]. Asian Journal of Technology Innovation, 2010, 18(2): 89-113.
- 【12】Sakazume, Y.. Is Japanese cell manufacturing a new system: a comparative study between Japanese cell manufacturing and cellular manufacturing?[J]. Journal of Japan Industrial Management Association, 2005, 55(6): 341-349.
- 【13】Murase, Y., Kaku, I., Yin, Y.. Theoretical analysis of converting assembly line to cellular manufacturing-How to select your manufacturing system[J]. Journal of Japan Industrial Management Association, 2006, 57(4): 314-322.
- 【14】Asano, S.. Internationalization of Japanese company and change of production system (in Japanese)[J]. Rikkyo Economic Review, 1997, 51(1): 29-55.
- 【15】Wemmerlöv, U., Hyer, N.. Procedures for the part family/machine group identification problem in cellular manufacturing[J]. Journal of Operations Management, 1986, 6(2): 125-147.
- 【16】Kimura, T., Yoshita, M.. Cellular manufacturing runs into trouble when nothing is done (in Japanese) [J]. Nikkei Monozukuri, 2004, 7: 38-61.
- 【17】Nonaka, I., Akira, K.. The Essence of Innovation (in Japanese) [M]. Japan: Nikkei BP, 2004.
- 【18】Isa, K., Tsuru, T.. Cell production innovation in Japan: toward a new model for Japanese manufacturing?[J]

Industrial Relations, 2002, 41(4): 548-578.

- 【19】 JSPMI. Cell manufacturing system and the innovation of manufacturing system, Research Report H9-9 (in Japanese)[R]. Tokyo: Japan Society for the Promotion of Machinery Industry, 1998.
- 【20】 Yin, Y., Kaku, I., Murase, Y., Yasuda, K.. *Seru* (cell) and reverse conversion: Part II. Homogenous Manufacturing Organizations[R]. Working paper of Yamagata University, 2008-E02. 2008a.
- 【21】 Sandberg, A.. Enriching Production: Perspectives on Volvo's Uddevalla Plant as An Alternative to Lean Production[M]. Aldershot: Avebury, 1995: 1-33.
- 【22】 Yamada, H., Kataoka T.. Unusual Production Revolution (in Japanese) [M]. Japan: NHK, 2001.
- 【23】 Selim, H.M., Abdel Aal, R.M.S., Mahdi, A.I.. Formation of machine groups and part families: a modified SLC method and comparative study[J]. Integrated Manufacturing Systems, 2003, 14(2): 123-137.
- 【24】 Defersha, F.M., Chen, M.Y.. A comprehensive mathematical model for the design of cellular manufacturing systems[J]. International Journal of Production Economics, 2006, 103: 767-783.
- 【25】 Chan, F.T.S., Lau, K.W., Chan, P.L.Y., Choy, K.L.. Two-stage approach for machine-part grouping and cell layout problems[J]. Robotics and Computer-Integrated Manufacturing, 2006, 22: 217-238.
- 【26】 Kim, C.O., Baek, J.G., Jun, Jin.. A machine cell formation algorithm for simultaneously minimising machine workload imbalances and inter-cell part movements[J]. The International Journal of Advanced Manufacturing Technology, 2005, 26(3): 268-275.
- 【27】 Al Kattan, I.. Workload balance of cells in designing of multiple cellular manufacturing systems[J]. Journal of Manufacturing Technology Management, 2005, 16(2): 178-196.
- 【28】 Albadawi, Z., Bashir, H.A., Chen, M.Y.. A mathematical approach for the formation of manufacturing cells[J]. Computers & Industrial Engineering, 2005, 48: 3-21.
- 【29】 Vitanov, V., Tjahjono, B., Marghalany, I.. A decision support tool to facilitate the design of cellular manufacturing layouts[J]. Computers & Industrial Engineering, 2007, 52: 380-403.
- 【30】 Wu, X.D., Chu, C.H., Wang, Y.F., Yan, W.L.. A genetic algorithm for cellular manufacturing design and layout[J]. European Journal of Operational Research, 2007, 181: 156-167.
- 【31】 Lee, S.D., Chen, Y.L.. A weighted approach for cellular manufacturing design: minimising inter-cell movement and balancing workload among duplicated machines[J]. International Journal of Production Research, 1997, 35: 1125-1146.
- 【32】 Papaioannou, G., Wilson, J.M.. The evolution of cell formation problem methodologies based on recent studies (1997-2008): Review and directions for future research[J]. European Journal of Operational Research, 2010, 206: 509-521.
- 【33】 Won, Y., Lee, K.C.. Modified p-median approach for efficient GT cell formation[J]. Computers and Industrial Engineering, 2004, 46: 495-510.
- 【34】 Chan, W.M., Chan, C.Y., Ip, W.H.. A heuristic algorithm for machine assignment in cellular layout[J].

- Computers and Industrial Engineering, 2003, 44: 49–73.
- 【35】 Kim, C.O., Baek, J.G., Baek, J.K.. A two-phase heuristic algorithm for cell formation problems considering alternative part routes and machine sequences[J]. International Journal of Production Research, 2004, 42 (18): 3911–3927.
- 【36】 Souilah, A.. Simulated annealing for manufacturing systems layout design[J]. European Journal of Operational Research, 1995, 82: 592–614.
- 【37】 Wu, T.H., Low, C., Wu, W.T.. A tabu search approach to the cell formation problem[J]. International Journal of Advanced Manufacturing Technology, 2004, 23: 916–924.
- 【38】 Solimanpur, M., Vrat, P., Shankar, R.. A multi-objective genetic algorithm approach to the design of cellular manufacturing systems[J]. International Journal of Production Research, 2004, 42 (7): 1419–1441.
- 【39】 Durán, O., Rodriguez, N., Consalter, L.A.. A PSO-based clustering algorithm for manufacturing cell design[C]. Proceedings of the 2008 Workshop on Knowledge Discovery and Data Mining, IEEE Computer Society. doi: 10.1109/WKDD.2008.1.
- 【40】 Yang, M.S., Yang, J.H.. Machine-part cell formation in group technology using a modified ART1 method[J]. European Journal of Operational Research, 2008, 188: 140–152.
- 【41】 Safaei, N., Saidi-Mehrabad, M., Tavakkoli-Moghaddam, R., Sassani, F.. A fuzzy programming approach for a cell formation problem with dynamic and uncertain conditions[J]. Fuzzy Sets and Systems, 2008, 159 (2): 215–236.
- 【42】 Nsakanda, A.L., Diaby, M., Price, W.L.. Hybrid genetic approach for solving large-scale capacitated cell formation problems with multiple routings[J]. European Journal of Operational Research, 2006, 171: 1051–1070.
- 【43】 James, T.L., Brown, E.C., Keeling, K.B.. A hybrid grouping genetic algorithm for the cell formation problem[J]. Computers and Operations Research, 2007, 34: 2059–2079.
- 【44】 Chandrasekaran, M.P., Rajagopalan, R.. An ideal seed non-hierarchical clustering algorithm for cellular manufacturing[J]. International Journal of Production Research, 1986, 24(2): 451–464.
- 【45】 Chandrasekaran, M.P., Rajagopalan, R.. ZODIAC: An algorithm for concurrent formation of part families and machine cells[J]. International Journal of Production Research, 1987, 25: 451–464.
- 【46】 Kumar, C.S., Chandrasekharan, M.P.. Grouping efficacy: A quantitative criterion for goodness of block diagonal forms of binary matrices in group technology[J]. International Journal of Production Research, 1990, 28(2): 233–243.
- 【47】 McCormick, W. T., Schweitzer, P. J., White, T. W.. Problem decomposition and data reorganization by a clustering technique[J]. Operations Research, 1972, 20(5): 993–1009.
- 【48】 Miltenburg, J., Zhang, W.. A comparative evaluation of nine well-known algorithms for solving the cell formation problem in group technology[J]. Journal of Operations Management, 1991, 10(1): 44–72.

- 【49】 Wisner, J.D.. A review of the order release policy research[J]. *International Journal of Operations and Production Management*, 1995, 15(6): 25-40.
- 【50】 Ahmed, I., Fisher, W.W.. Due date assignment, job order release, and sequencing interaction in job shop scheduling[J]. *Decision Science*, 1992, 23(3): 633-647.
- 【51】 Kanet, J.J.. Load-limited order release in job shop scheduling systems[J]. *Journal of Operations Management*, 1988, 7(3-4): 44-58.
- 【52】 Park, P.S., Bobrowski, P.M.. Job release and labor flexibility in a dual resource constrained job shop[J]. *International Journal of Operations Management*, 1989, 8(3): 230-249.
- 【53】 Cigolini, R., Perona, M., Portioli, A.. Comparison of order review and release techniques in a dynamic and uncertain job shop environment[J]. *International Journal of Production Research*, 1998, 36(11): 2931-2951.
- 【54】 Süer, G.A., Saiz, M., Gonzalez, W.. Evaluation of manufacturing cell loading rules of independent cells[J]. *International Journal of Production Research*, 1999, 37(15): 3445-3468.
- 【55】 Mahmoodi, F., Dooley, K.J., Starr, P.J.. An evaluation of order releasing and due date assignment heuristics in a cellular manufacturing system[J]. *Journal of Operations Management*, 1990, 9(4): 548-573.
- 【56】 Süer, G.A., Ortega, M.. A machine level based-similarity coefficient for forming manufacturing cells[J]. *Computers and Industrial Engineering*, 1994, 27(1-4): 67-70.
- 【57】 Davis, D.J., Mabert, V.A.. Order dispatching and labor assignment in cellular manufacturing systems[J]. *Decision Sciences*, 2000, 31(4): 745-771.
- 【58】 Süer, G.A., Cosner, J., Patten, A.. Models for cell loading and product sequencing in labor-intensive cells[J]. *Computers and Industrial Engineering*, 2009a, 56: 97-105.
- 【59】 Greene, T.J., Sadowski, R.P.. Cellular manufacturing control[J]. *Journal of Manufacturing Systems*, 1984, 4: 137-145.
- 【60】 Morris, J.S., Tersine, R.J.. A comparison of cell loading practices in group technology[J]. *Journal of Manufacturing and Operations Management*, 1989, 2: 299-313.
- 【61】 Shafer, S.M., Charnes, J.M.. A simulation analyses of factors influencing loading practices in cellular manufacturing[J]. *International Journal of Production Research*, 1995, 33(1): 279-290.
- 【62】 Espino, M.. Scheduling in cellular manufacturing with interruptions in the process flow[D]. Unpublished M.S Thesis, University of Puerto Rico-Mayaguez, Puerto Rico, 1991.
- 【63】 Süer, G.A., Saiz, M., Dagli, C., Gonzalez, W.. Manufacturing Cell Loading Rules and Algorithms for Connected Cells[M]. *Planning, Design, and Analysis of Cellular Manufacturing Systems*, Amsterdam: Elsevier, 1995.
- 【64】 Süer, G.A., Saiz, M., Gonzalez, W.. Evaluation of manufacturing cell loading rules of independent cells[J]. *International Journal of Production Research*, 1999, 37(15): 3445-3468.

- 【65】Süer, G.A., Subramanian, A., Huang, J.. Heuristic procedures and mathematical models for cell loading and scheduling in a shoe manufacturing company[J]. Computers and Industrial Engineering, 2009b, 56: 462-475.
- 【66】Süer, G.A., Dagli, C.. Intra-cell manpower transfers and cell loading in labor-intensive manufacturing cells[J]. Computers and Industrial Engineering, 2005, 48: 643-655.
- 【67】Süer, G.A., Vazquez, R., Cortes, M.. A hybrid approach of genetic algorithms and local optimizers in cell loading[J]. Computers & Industrial Engineering, 2005, 48: 625-641.
- 【68】Akturk, M.S., Wilson, G.R.. A hierarchical model for the cell loading problem of cellular manufacturing systems[J]. International Journal of Production Research, 1998, 36(7): 2005-2023.
- 【69】Babayigit, C.. Genetic algorithms and mathematical models in manpower allocation and cell loading problem[D]. Unpublished M.S Thesis, The Ohio University, 2003.
- 【70】Yarimoglu, F.. Cell loading and product sequencing subject to manpower restrictions in synchronized manufacturing cells[D]. Unpublished M.S Thesis, The Ohio University, 2009.
- 【71】Hax, A.C., Meal, H.C.. Hierarchical Integration of Production Planning and Scheduling[M]. In Studies in Management Science, M.A. Geisler(ed), Amsterdam: North-holland, 1975.
- 【72】Bellman, R.E., Zadeh, L.A.. Decision making in a fuzzy environment[J]. Management Science, 1970, 17(4): 141-164.
- 【73】Zimmermann, H.J.. Description and optimization of fuzzy systems[J]. International Journal of General Systems, 1976, 2(4): 209-215.
- 【74】Süer, G.A., Arikan, F., Babayigit, G. Bi-objective cell loading problem with non-zero setup times with fuzzy aspiration levels in labour intensive manufacturing cells[J]. International Journal of Production Research, 2008, 46(2): 371-404.
- 【75】Süer, G.A., Arikan, F., Babayigit, G. Effects of different fuzzy operators on fuzzy bi-objective cell loading problem in labor-intensive manufacturing cells[J]. Computers & Industrial Engineering, 2009c, 56: 476-488.
- 【76】Holland, J.H.. Adaptation in Natural and Artificial Systems[M]. MI: University of Michigan Press, 1975.
- 【77】Mahdavi, I., Paydar, M.M., Solimanpur, M., Heidarzade, A.. Genetic algorithm approach for solving a cell formation problem in cellular manufacturing[J]. Expert Systems with Applications, 2009, 36: 6598-6604.
- 【78】Falkenauer, E.. Genetic Algorithms for Grouping Problems[M]. NY: Wiley, 1998.
- 【79】Michalewicz, Z.. Genetic Algorithms+Data Structures=Evolution Programs[M]. AI Series, Springer, New York, 1996.
- 【80】Bolat, A., Al-Harkan, I., Al-Harbi, B.. Flow-shop scheduling for three serial stations with the last two duplicate[J]. Computers & Operations Research, 2005, 32: 647-667.

- 【81】Sevauz, M., Dauzere-Peres, S.. Genetic algorithms to minimize the weighted number of late jobs on a single machine[J]. European Journal of Operational Research, 2003, 151: 296-306.
- 【82】Onwubolu, G.C., Mutingi, M.. A genetic algorithm approach to cellular manufacturing systems[J]. Computers & Industrial Engineering, 2001, 39: 125-144.
- 【83】James, T.L., Brown, E.C., Keeling, K.B.. A hybrid grouping genetic algorithm for the cell formation problem[J]. Computers & Operations Research, 2007, 34: 2059-2079.
- 【84】Pitombeira Neto, A.R., Gonçalves Filho, E.V.. A simulation-based evolutionary multiobjective approach to manufacturing cell formation[J]. Computers & Industrial Engineering, 2010, 59: 64-74.
- 【85】Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.. A fast and elitist multi-objective genetic algorithm: NSGA-II[J]. IEEE Transactions Evolutionary Computation, 2002, 6(2): 182-197.
- 【86】Pearl, J.. Heuristic: Intelligent Search Strategies for Computer Problem Solving[M]. New York, Addison-Wesley, 1983.
- 【87】刘晨光, 廉洁, 李文娟, 殷勇. 日本式单元化生产——生产方式在日本的最新发展形态[J]. 管理评论, 2010, 22: 93-103.
- 【88】Vakharia, A.J., Kaku, B.K.. Redesigning a cellular manufacturing system to handle long-term demand changes: a methodology and investigation[J]. Decision Sciences, 1993, 24(5): 909-929.
- 【89】Harlahakis, G., Ioannou, G., Minis, I., Nagi, R.. Manufacturing cell formation under random product demand[J]. International Journal of Production Research, 1994, 32(1): 47-64.
- 【90】Yin, Y., Kaku, I., Stecke, K.E.. The evolution of seru production systems throughout Canon[J]. Operations Management Education Review, 2008b, 2: 27-40
- 【91】Burbidge, J.L.. Production flow analysis[J]. Production Engineer, 1963, 42(12): 742-752.
- 【92】Joines, J.A., Culbreth, C.T., King, R.E.. Manufacturing cell design: An integer programming model employing genetic algorithms[J]. IEEE Transactions, 1996, 28(1): 69-85.
- 【93】Gonçalves Filho, E.V., Tiberti, A.J.. A group genetic algorithm for the machine cell formation problem[J]. International Journal of Production Economics, 2006, 102: 1-21.
- 【94】Gen, M., Cheng, R.. Interval programming using genetic algorithms[C]. Proceedings of the Sixth International Symposium on Robotics and Manufacturing, Montpellier, France, 1996a.
- 【95】Gen, M., Cheng, R.. A survey of penalty techniques in genetic algorithms[C]. Evolutionary Computation, IEEE International Conference on, 1996b, 804-809.
- 【96】Schaffer, J.D.. Multiple objective optimization with vector evaluated genetic algorithms and their applications[C]. Proceedings of the First International Conference on Genetic Algorithms, 1985, 93-100.
- 【97】Brown, E., Sumichrast, R.. CF-GGA: a grouping genetic algorithm for the cell formation problem[J]. International Journal of Production Research, 2001, 39(16): 3651-3669.

硕士期间发表论文及所获奖励

发表论文:

- 【1】Liu, C.G., Yin, Y., Yasuda. K., Lian, J.. A heuristic algorithm for cell formation problems with consideration of multiple production factors[J]. The International Journal of Advanced Manufacturing Technology, 2010, 46: 1201-1213. (论文被 SCI 检索收录, 检索号 552XI)
- 【2】刘晨光,廉洁,李文娟,殷勇.日本式单元化生产——生产方式在日本的最新发展形态[J].管理评论, 2010, 22: 93-103. (论文被 CSSCI 检索收录, 检索号 11C1172010050013; 并被人大复印资料 F31《企业管理研究》2010 年第 8 期全文转载)
- 【3】Liu, C.G., Lian, J., Yin, Y., Li, W.J.. Seru seisan — an innovation of the production management mode in Japan[J]. Asian Journal of Technology Innovation, 2010, 18(2): 89-113. (论文被 SSCI 检索收录, 检索号 703CV)
- 【4】Zhu, L., Lian, J.. Worker flexibility oriented cross-training[C]. IEEE Third International Conference on Education Technology and Training (ETT), 2010.
- 【5】Liu, C.G., Li, W.J., Lian, J., Bai, Y.F., Yin, Y.. A mathematical model for reconstructing the conventional assembly system[C]. Industrial Engineering and Engineering Management (IE&EM), 2011. (论文被 EI 检索收录, 检索号 20114514486852)
- 【6】刘晨光,白玉芳,李文娟,廉洁,殷勇.两类市场结构下企业生产方式选择决策[J].系统工程理论与实践, 2012, 32(1): 49-59. (论文被 EI 检索收录, 检索号 20121114863150)
- 【7】Liu, C.G., Li, W.J., Lian, J., Yin, Y.. Reconfiguration of assembly systems: from conveyor assembly line to seru[J]. Journal of Manufacturing Systems. (论文已录用)

所获科研奖励:

- (1) 2011 年陕西高等学校人文社会科学研究优秀成果二等奖;
- (2) 陕西省第十次哲学社会科学优秀成果三等奖;

所获荣誉:

- (1) 2010 学年西安理工大学研究生学术年会“优秀论文奖”;
- (2) 2011 学年西安理工大学研究生学术年会“优秀论文奖”;
- (3) 2009-2010 学年校三等奖学金;
- (4) 2010-2011 学年东方电气奖学金;
- (5) 2009-2010 学年三好学生荣誉称号;
- (6) 2010-2011 学年三好学生标兵荣誉称号。

面向Seru生产的生产单元构建与生产任务分配决策研究



作者：[廉洁](#)
学位授予单位：[西安理工大学](#)

引用本文格式：[廉洁](#) [面向Seru生产的生产单元构建与生产任务分配决策研究](#)[学位论文]硕士 2012