第50卷第4期
2010年7月

大连理工大学学报
Journal of Dalian University of Technology

Vol. 50, No. 4
July 2010

文章编号：1000-8608(2010)04-0535-05

含装配约束的作业调度问题及遗传解码算法

王林平[1,2]， 贾振元*[1]， 王福吉[1]， 孟凡斌[1]

（1. 大连理工大学 精密与特种加工教育部重点实验室，辽宁 大连 116024；
2. 烟台大学 机电汽车工程学院，山东 烟台 264005）

摘要： 传统上加工与装配是分开研究的，而且很少涉及作业装配问题. 为此，描述了一种包含加工与装配的作业调度问题. 利用 FT10 基准调度问题的工序数据构造了调度问题及其产品结构. 初步进行了以经典作业调度问题(JSSP)为对照的研究. 采用基于字符串直接解码的遗传算法，解决了染色体对应不可行调度解的问题. 仿真实验验证了遗传解码算法的可行性和有效性，而且进一步表明装配约束改变了 JSSP；近优值 2 046 也优于关键路径规则算法的结果.

关键词： 装配约束；加工；遗传算法；解码；作业调度问题

中图分类号： TP301　　**文献标志码：** A

0 引 言

作业调度问题(job-shop scheduling problem, JSSP)是一类满足顺序约束要求的任务配置和资源分配问题，也是最困难的组合优化问题之一，其研究已经有几十年的历史. 国内外对产品加工调度问题的研究较早，而且取得了大量的研究成果，已将众多理论或方法如动态规划、分枝定界法、神经网络法和遗传算法等应用其研究[1]，研究和开发了许多新的调度算法[2~5]. 然而，由于经典 JSSP 过于抽象，通常不能较好地应用于实际生产调度，人们也研究提出了多种推广：与工序有关的准备时间、柔性工艺路线、可重入等. 但是发表的绝大多数论文并未考虑将装配约束纳入经典的 JSSP，例如涉及具有复杂产品结构的作业调度[6]. 有关产品装配调度问题的研究成果，主要集中在大批量流水线(flow-shop)的产品装配上，近年来逐步扩展到批量混流装配线调度的研究[7].

另一方面，生产问题研究，大多偏向以自动化程度较高的加工系统如流水线或 FMS 为背景，而很少顾及以手工作业为主的装配系统[8]. 对于小批量、高复杂度产品作业装配问题，研究成果较少. 还有，大型或复杂产品的装配过程有其特殊要求，对大型产品的装配过程进行规划时不仅要考虑装配的顺序和最终的结果，还要考虑到借以进行装配的条件如在制品可用性等. 在多品种、小批量生产模式下，除混流装配线外，采用工作台以作业方式组织装配也是一个重要的选择，具有广泛的实际背景. 本文以经典 JSSP 问题为对照，研究一种包括加工与装配约束的作业调度问题，在这里将这种包括全部制造过程的生产作业调度问题称为全生产作业调度问题(complete job-shop scheduling problem, CJSSP)，实际上也是企业制造链的调度问题，已有的少量成果主要是基于规则的调度研究[9~11]. 本文在分析和描述问题的基础上，用复合结构的信息进行基于工序的编码，应用选择字符串编码方法直接求解，设计改进的遗传算法.

1 CJSSP 描述

产品的加工过程可以分为加工作业和装配作业，装配作业通常非常复杂而且有很大的纵向深度，要包括很多的总成、自制件和外购件. 图 1 列举了一种典型的相对简单的产品树状结构，其中根节点表示最终的产品，叶节点表示组成产品的自制件和外购件，其他节点表示装配过程中出现的总成. 为了表示各个节点以及它们之间的区别，下面对节点进行如下定义.

收稿日期： 2008-01-28； **修回日期：** 2010-03-10.
基金项目： “八六三”国家高技术研究发展计划资助项目(2003AA414022).
作者简介： 王林平(1963-)，男，博士，E-mail：qutinwlp@yahoo.com.cn；贾振元*(1963-)，男，博士，教授，博士生导师，E-mail：jzyxy@dlut.edu.cn.

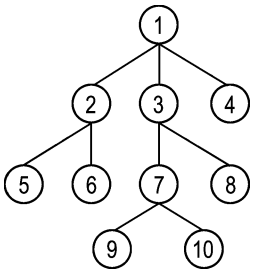

图 1 典型的产品结构

Fig. 1 An example of typical product structure

CJSSP 涉及多种物料类型、产品生产工艺、产品装配结构. 为使描述更清晰,定义装配体(constituent)概念:不加区分地表示组成产品的所有物料(item),包括零件(part)、部件(component)和产品(product),本文需要明确某一种物料时给出相应说明. 装配作业调度与经典的加工作业调度问题不同之处在于,除 JSSP 具有的所有约束外,在装配过程中还存在装配物料之间的约束:所有的父项装配体,只有在其所有直接子项装配体都已经加工完毕后,方可开始装配生产. 为了唯一确定一个装配体在产品结构中的位置,表示出产品加工与装配工艺,在描述产品装配结构的装配体编号(item number)基础上建立标识体系. 编号方法为从 0 层的最终装配体(产品)开始,自上而下逐层标识数字号码 1,2,3,…;对位于同一层的装配体,从左至右标号. 图 1 给出了一个典型的简单产品装配结构,标号 1 是最终的装配体,即产品;2 是总装之前的装配体,即部件;装配体 3、4、5、6 均为零件. 图 1 中,所有装配体的加工或装配工序附属于各自的节点.

从上面的叙述可以看出,装配体编号不是生产管理信息系统中的物料分类编码(item code). 通过编号与物料编码之间的关系数据库表,可以获取各种物料的基本数据:物料名称、类型、工艺路线及其准备和加工时间等数据.

加工与装配作业调度的基本问题可描述为 p 个产品,每个产品有不相等的 n 个装配体,每个零件有不相等的 o 个工序,在 m 台机器上加工,已知 p 个产品的结构、各工序的制造时间和各装配体在各机器上的制造次序约束,要求确定与产品结构和工艺条件相容的各机器上所有装配体的制造开始时间或完成时间或制造次序,使制造性能达到最优.

2 遗传算法设计

相对于经典的 JSSP,增加的装配约束使得大多数编码方法都会出现不可行解染色体,多产品的存在也增加了遗传操作的难度. 遗传算法采用基于工序的等长编码,以完全随机方式产生初始种群,用轮盘赌进行遗传选择,使用精英保留策略. 适应值是最大产品完工时间的简单变换,交叉算子采用部分映射交叉方式,变异算子采用逆序操作. 因为遗传操作后的新染色体基本都是不可行解,采用了一种基于字符串的解码方式将遗传操作后的每一个染色体都解码为唯一的可行解. 对编码和解码方法做进一步说明如下.

2.1 产品结构描述与编码设计

下文以单产品问题为例,也可推广至多产品调度问题. 为了唯一确定装配体的位置和相互关系,对每个装配体给出一个产品结构标识符(product structure identifier,PSI),以明确所有装配体及其工序之间的关系,如图 2 所示,C、E、Z、B 分别表示装配体的物料分类编码为产品、部件、自制零件、标准件. 根据物料编码,可以获得该物料的各种工艺参数,例如工序及其加工时间、加工设备等.

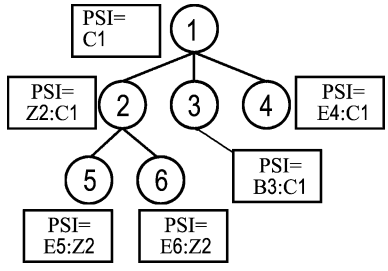

图 2 产品结构标识

Fig. 2 Product structure identifier (PSI) sample

遗传算法采用基于复合工序信息的编码方式,基因在染色体中的顺序表示工序排产的次序. 染色体的每个基因表示一个工序,它含有两层数据:该工序所属装配体的编号、工序号. 在染色体编码中,将表示节点类型的字母去除以有利于提高算法处理速度,为了在染色体中便于识别,统一加上了标识符 P,具体结构参见下面的变异操作示意图 3.

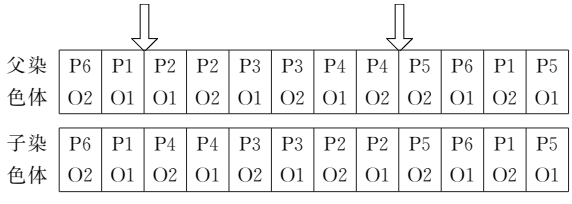

图 3 逆序变异操作

Fig. 3 Inversion mutation (IM) of the sample

2.2 交叉与变异操作

在初始种群中,按照染色体的适应度值折算

的概率选取染色体进行交叉或者变异操作，适应度大的染色体被选择的概率较大，从而保证优势个体在种群中以更大的概率存活.

交叉是主要的遗传操作，新染色体大部分由交叉获得，交叉操作能够有效地保留父项信息，本文交叉算子采用部分映射交叉方式.

变异操作的目的是通过随机改变染色体的某些基因来引入新个体，增加种群的多样性，本文的变异算子采用逆序操作，如图 3 所示.

3　遗传解码设计

装配约束的引入，使得装配工序必须在依附于子项节点的全部工序完成之后方可进行；应用于经典 JSSP 的基于工序编码的遗传算法，应用于 CJSSP 时就无法直接得到可行的调度方案. 在下面的叙述中，本文将这种按通常解码算法对应不可行调度方案的染色体称为不可行染色体. 因某种编码或遗传操作而产生不可行染色体时，遗传算法中可以有两种处理办法：一是设定不可行染色体的适应值为一个很小的数甚至为零，在遗传操作过程中逐渐将其从种群中去除；二是修正不可行染色体为可行染色体. CJSSP 中不可行调度方案占编码空间的比例较大，粗略估计会达到一半以上，第一种办法不再适用. 当然，一般来说将不可行染色体转化为可行染色体的过程并不容易.

3.1　构造选择解码字符串

通过研究分析，本文将确定装配体相对位置的装配结构标识符进行改造，构造了一种选择解码字符串（selective decoding string，SDS），将不可行染色体直接解码为唯一可行调度，以避开复杂的修正过程. SDS 由 PSI 和表示工件工序数 NOI（number of operations）的字符串合成，用于表示解码过程中装配体之间的先后顺序，如图 4 所示，①列表示装配体编号，②列是紧后父项装配体编号，③行是所属产品代码，NOI 列表示①列对应该装配体的工序数量. 图中的圈定行，表示属于 1 产品的 4 号工件的全部 1 个工序都完成后，1 工件才可以开始进行加工.

3.2　解码算法设计

首先进行 SDS 初始化：确定待添加节点对应的产品代码，然后逐层从小到大、同一层次自左而右找到装配体节点及其对应的父节点，将其编号填入第①和第②列，最后把查询到的依附于该节点装配体的工序数填入 NOI 列中. 循环进行添加操作直至将所有节点装配体的上述数据全部填入 SDS 中.

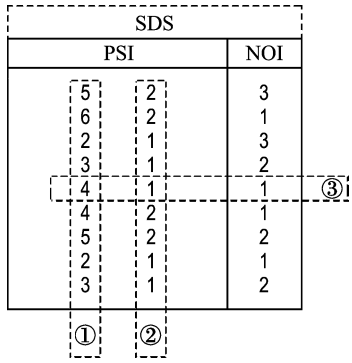

图 4　选择解码字符串

Fig. 4　Selective decoding string(SDS)

SDS 解码步骤如图 5 所示，算法描述如下：

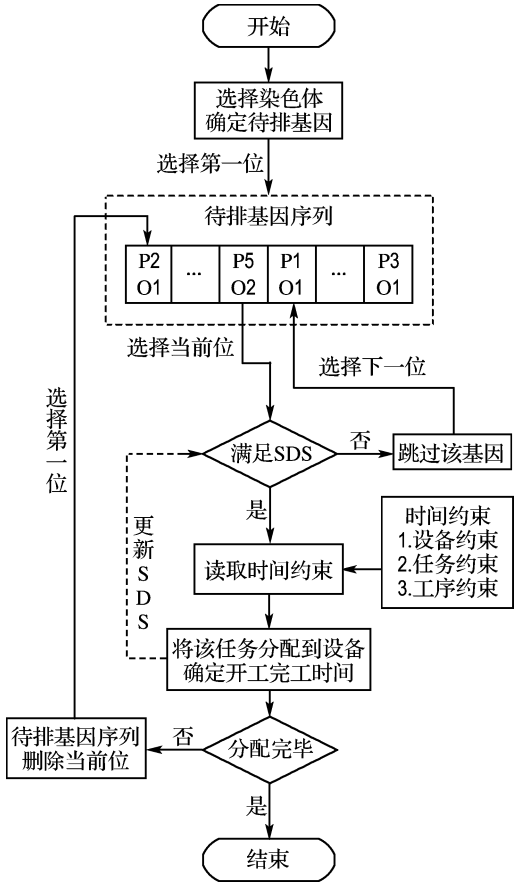

图 5　遗传解码步骤

Fig. 5　Genetic decoding procedure

(1)将已经初始化的待解码染色体 SDS 存放在数组 Chr()中（为了与染色体相区别，将该数组称为待解码基因序列）. 设染色体左数第一位基因为当前基因.

(2)从待解码基因序列中取出当前基因，根据当前 SDS 的数值判断其是否可以排产. 如可排，则跳转至(3)；不可排，则当前位后移一位，进行(2)；如果已至末尾，则将当前左数第一位基因设

为当前基因.重复进行,直至找到可排产的基因.

(3)将该基因从待解码基因序列移至 gene()中,在数组 num 中将该基因对应工件的工序数减 1.

(4)查找基因对应工序的加工时间 time().

(5)根据设备 machine_time()、任务 process_time()、工序 task_time()三个约束参数,确定该基因对应工序最早的派工时间.计算该工序开工时间 sta_time()和完工时间 end_time().

(6)根据开工和完工时间确定该工序在甘特图中的坐标,画出该工序的解码甘特图块.

(7)依据新工序的调度结果,更新派工约束 machine_time()、process_time()和 task_time().

(8)如果 SDS 工序数列 num()中该工件对应的工序数为 0,则从 SDS 中删除该行记录.

(9)如解码基因序列 Chr()不为空,则设置其左数第一位基因为当前基因,跳转至(2);如果 Chr()为空,染色体解码结束,输出派工结果甘特图.

4 仿真结果及分析

对于 CJSSP,还未曾有人提出类似经典 JSSP 调度问题的算例[1,3];已经发表的相关文献中,也缺乏完整的问题说明.部分原因可能是这类问题没有受到应有的重视,再加上数据规模本身也比较大.因此,作者以著名的 FT10 问题作为基础,构建了本文的调度问题算例 FT10C,其产品数据结构如图 1 所示,具体的工序加工时间数据与 FT10 完全相同,见文献[3]附录 1,产品的节点标号 1,2,…,10,分别对应于 FT10 问题中的 0～9 号工件.对于多个产品的加工与装配作业调度问题,可以将多个产品合成为一个虚拟的大产品,其装配工序数为 1,工时为 0.算法用 VB.NET 进行开发,运行计算机的 CPU 为 Celeron,内存 512 MB,主频 2.0 GB.

以经验确定的遗传算法参数为种群 100,交叉概率 0.6,变异概率 0.01,所有重复实验均采用.实验重复进行 10 次,每次遗传 200 代,全部得到了问题的可行解.图 6 是 10 次实验的最佳与平均完工时间 t 随遗传代数 n 变化的折线图.10 次实验得到最优值 2 046 是第 1 次实验中经过 170 代进化的结果,对应调度方案甘特图为图 7,显示调度方案是可行的,没有违反加工和装配工艺的情况发生.这说明设计的遗传解码方法是可行的.

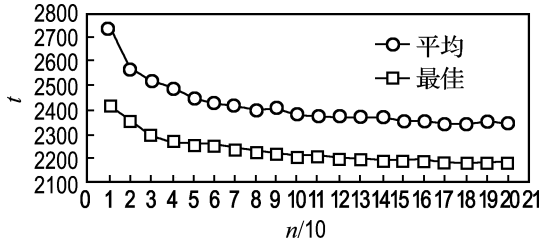

图 6 仿真结果

Fig. 6 Simulation results

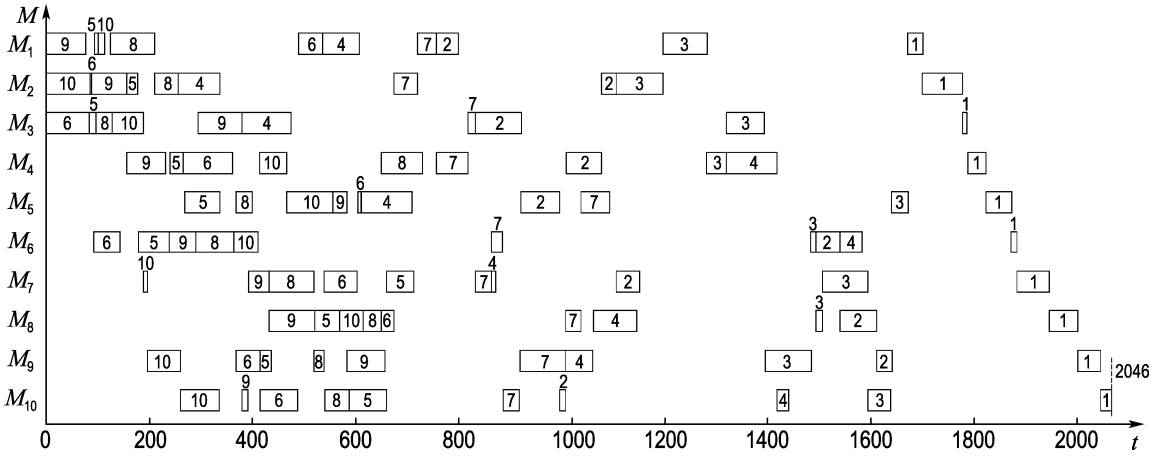

图 7 近优值 2 046 对应的机器甘特图

Fig. 7 The machine-oriented Gantt chart for the near-optimal value 2 046

虽然难于得到问题的最优解,因而不能证明 2 046 就是问题的最优解,但是,利用网络规划中关键路线的概念,并假设设备有无限能力,即不受设备能力约束的情况,可以计算得到此问题的最短路线为 1 976.使用优化目标为最大完工时间的规则算法,在设备的有限能力约束下,利用基于关键路线概念的 CP 调度规则[6],可得到完工时间为 2 379.

遗传算法求得的最好解与 1 976 的差为 70,相对误差为 3.54%,但是大大优越于基于 CP 调度规则

的结果 2 379. 再考虑到此问题 100 个工序的平均加工时间为 51.09，2 046 已经很接近此问题的最优解.

5 结　语

JSSP 的难解特性与广泛的工程背景人们已经熟知. 本文描述了一类比 JSSP 更接近实际生产的复杂调度问题：增加了多层装配约束的 CJSSP. 用遗传算法求解问题时，往往碰到处置染色体的解码为不可行解的问题. 不可行染色体数量少时，可以考虑用适应函数罚值或直接舍弃的办法处理. 面对大比例不可行染色体时，须采用复杂的修补方法将之转换为可行解.

本文根据包含问题装配约束的选择解码字符串直接进行解码. 仿真结果得到了 FT10C 的近似优化解为 2 046，接近于无限能力下的更小的下限值 1 976，也比关键路线规则下的 2 379 好了许多；当然比 FT10 的完工时间最优值 930[12] 已经大了许多，充分说明了排产的困难.

参考文献：

[1] JAIN A S, MEERAN S. Deterministic job-shop scheduling: Past, present and future [J]. **European Journal of Operational Research**, 1999, **113**(2): 390-434

[2] BECK J C. Solution-guided multi-point constructive search for job shop scheduling [J]. **Journal of Artificial Intelligence Research**, 2007, **29**: 49-77

[3] 王　凌. 车间调度及其遗传算法[M]. 北京：清华大学出版社，2003

[4] ZHANG C Y, LI P G, GUAN Z L, *et al*. A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem [J]. **Computers & Operations Research**, 2007, **34**(11): 3229-3242

[5] 吴　怡，刘　民，吴　澄. JSSP 基本约束特点分析及调度算法[J]. 清华大学学报(自然科学版)，2004，**44**(10): 1380-1383

[6] PINEDO M L. **Planning and Scheduling in Manufacturing and Services: Springer Series in Operations Research and Financial Engineering** [M]. New York: Springer, 2005

[7] 曾洪鑫，宾鸿赞，张　芬. 多品种装配顺序的一种改进遗传算法[J]. 华中科技大学学报(自然科学版)，2006，**34**(3): 39-41

[8] 吴　锋，刘文煌. 手工装配作业下的一种调度策略与控制方法[J]. 清华大学学报(自然科学版)，1998，**38**(2): 58-61

[9] NATARAJAN K, MOHANASUNDARAM K M, SHOBAN BABU B, *et al*. Performance evaluation of priority dispatching rules in multi-level assembly job shops with jobs having weights for flowtime and tardiness [J]. **International Journal of Advanced Manufacturing Technology**, 2007, **31**(7-8): 751-761

[10] CHENG T C E. Optimal due-date assignment in an assembly shop [J]. **International Journal of Operations & Production Management**, 1994, **14**(2): 31-42

[11] HICKS C, PONGCHAROEN P. Dispatching rules for production scheduling in the capital goods industry [J]. **International Journal of Production Economics**, 2006, **104**(1): 154-163

[12] CARLIER J, PINSON E. An algorithm for solving the job-shop problem [J]. **Management Science**, 1989, **35**(2): 164-176

Job-shop scheduling problem incorporating assembly constraint and its solution using genetic decoding approach

WANG Lin-ping[1,2], JIA Zhen-yuan[*1], WANG Fu-ji[1], MENG Fan-bin[1]

(1. Key Laboratory for Precision & Non-traditional Machining Technology of Ministry of Education, Dalian University of Technology, Dalian 116024, China;
2. School of Electromechanical Automobile Engineering, Yantai University, Yantai 264005, China)

Abstract: Scheduling for fabrication and assembly operations is researched independently. So far there are relatively less scheduling studies concerning job-shop assembly operation. To solve this problem, a job-shop scheduling problem (JSSP) including both fabrication and assembly constraint operations is described. The problem instance is established on the basis of benchmark FT10 operation data. It was preliminarily investigated by contrasting it to classical JSSP. A selective decoding string was constructed to overcome the unfeasible solutions resulting from chromosomes. Simulation tests indicate that the genetic decoding approach is both feasible and effective. Furthermore, they demonstrate that the assembly constraints have made a great difference in JSSP. Moreover, a near-optimal makespan of 2 046 is obtained, which is better than the result of CP dispatching rule.

Key words: assembly constraint; fabrication; genetic algorithm; decoding; job-shop scheduling problem (JSSP)