

A Genetic Algorithm for Solving Dynamic Scheduling Problems in Distributed Manufacturing Systems

Yanhong Wang¹, Lixin Yan¹, Hongyu Zhu¹, Chaowan Yin²

¹) Department of Information Sciences and Engineering, Shenyang University of Technology, Shenyang 110023
(E-mail: wangyh@sut.edu.cn)

²) Shenyang Institute of Automation, Chinese Academy of Science, Shenyang 110016
(E-mail: ycw@sia.ac.cn)

Abstract—It have been proven that the distributed manufacturing system, if managed properly, can enable enterprises to reduce manufacturing costs, increase products quality and make better use of manufacturing resources. However, the dynamic scheduling in distributed manufacturing environments can be much more complex than that in the single integrated enterprise cases. In this paper, a distributed scheduling method is developed, which is composed of an iterative coordination mechanism and a modified genetic algorithm. The complicated scheduling problem is divided into several sub-problems to make the problem easier. The scheduling objective is to achieve a multiple performance index, i.e. minimizing the manufacturing cost and meeting the due date. The capability of the proposed method has been tested with satisfactory results through several numerical experiments.

Keywords—dynamic scheduling, genetic algorithms, distributed manufacturing system

用遗传算法求解分布式制造系统动态调度问题

王艳红¹ 闫丽新¹ 祝洪宇¹ 尹朝万²

¹) 沈阳工业大学信息科学与工程学院 沈阳 110023

²) 中国科学院沈阳自动化研究所 沈阳 110016

摘要 在分布式制造环境下, 调度问题的规模更大, 所涉及的对象更复杂, 系统的动态性更强, 使得调度问题求解的难度进一步增加。本文提出了一种求解分布式制造系统动态调度的改进遗传算法。为此首先简要描述了分布式制造调度问题模型, 建立了用于系统各分布制造单元间彼此协调的分布调度机制。在此基础上, 设计实现了分布调度遗传算法, 该算法能够完成工件制造任务在分布制造单元及其加工机器间的优化调度, 并使调度结果在满足交货期的同时, 取得最小制造成本。最后, 给出了应用此方法完成分布式动态调度的仿真算例, 仿真结果表明所提出的调度策略是可行、有效的。

关键词 动态调度, 遗传算法, 分布式制造系统

1. 引言

分布式制造系统可以超越物理空间和时间的约束, 充分利用分散在各地的人力、机器设备、原材料等资源, 通过有效的资源优化整合与协调, 迅速响应市场需求, 有效地提高产品产量、降低生产成本。然而, 在分布式制造环境下制造系统的动态性、不确定性和非预见性特征更加明

显, 对生产系统的管理与控制技术, 包括生产调度方法提出了更多的挑战。

制造调度问题是一类困难的组合优化问题, 也是典型的 NP 难题。在分布式制造环境下, 调度系统既要处理好作业与资源的优化匹配问题, 还要处理好异地制造资源的协同问题; 既要具有传统意义下的优化能力, 还要具有敏捷反应及适应能力, 因而调度问题求解的难度进一步增加。关于传统车间调度问题的研究已经有很多年的历史, 人们提出多种方法, 取得了很多研究成果。1985 年 Davis 首次

辽宁省自然科学基金项目 (资助号: 20042031)

辽宁省高校科研基金项目 (资助号: 2004D036)

将遗传算法用于求解调度问题，此后人们从单目标的、多目标的；有约束的、无约束的等各类调度问题着手进行遗传算法应用研究，有单纯采用遗传算法的，有与其他方法集成的。但是，用遗传算法解决分布式调度问题的研究还比较少。文献[1]提出一种改进的遗传算法，提出了用两步编码方法来产生表示分布式调度任务的染色体，并通过设计一次交叉和两次变异的遗传操作来改变工件的加工工序序列和所选择的工厂，从而解决分布式调度问题。然而，该算法解决的主要还是静态调度问题，没有考虑调度任务动态变化的情况。文献[2]给出了基于遗传算法的网络化制造系统协调调度框架，设计了编码方法，并通过 Agent 实现异地调度系统之间的协调通讯。

本文将分布式制造系统调度问题域分解为制造单元之间的制造任务协同优化调度、各个制造单元内部工件制造过程优化调度以及在工件加工过程中制造单元之间的协调等调度子问题，给出了调度过程的描述，并重点讨论制造单元间任务分配的协调调度策略及其遗传算法实现。

2. 问题的描述

分布式制造系统由分布在不同物理空间的制造实体组成，这些制造实体可能是传统意义上的独立的工厂或车间，它们在网络的支持下，通过彼此协调与合作，实现共赢的利益目标。制造系统的组成既可是相对稳定的，如跨国公司，也可是暂时的，如虚拟联盟企业。在传统制造环境中，某个制造实体仅依据自身的情况完成调度决策，而在分布式制造环境中，要实现多约束、多目标条件下，作业任务在多个分布、异地、自治的制造实体中的调度，问题显然复杂得多。本质上来讲，分布式制造系统的调度问题仍是一个作业与资源的优化分配问题，然而在分布式制造环境下，调度问题的规模更大，系统所涉及对象更复杂、信息的易变性和不确定性更强，调度中的信息跟踪与交互、各制造实体的调度协调与优化等问题更突出^[4]，调度质量的目标也不再仅仅局限于满足某些常规的运作性能指标，而转向追求最小化的整体运作成本、最大化的客户满意度等综合性性能指标，同时，为了适应动态变化的市场，特别强调调度的敏捷性。

2.1 几个定义

分布式制造系统由若干独立的具有加工能力的工厂或车间等制造实体组成，为下文叙述方便，给出相关定义。

定义 1 制造单元 $Cell_i$ ：组成分布式制造系统的、彼此独立的具有加工能力的工厂或车间等制造实体。一个分布式制造系统包括多个 $Cell_i$ ，它们可以动态加入或退出。

定义 2 盟主 $Leader_k$ ：制造调度任务的发起者和组织

者，并负责其它制造单元制造任务调度结果的评价和最终分配。在分布式系统中，根据系统体制的不同， $Leader_k$ 可以是固定的（比如在跨国公司中），也可以是临时的（比如在虚拟企业中）。在后一情况中，其生命周期随任务的产生而开始，随任务的结束而终结。系统允许多个 $Leader_k$ 同时存在，

定义 3 作业任务 Job_j ：直接面向制造单元的可执行的制造任务，是制造订单中的多品种制造任务或大型复杂产品，经基于时间连贯性或加工一致性等原则完成分解、重组后形成的。每个 Job_j 通过多种加工（多道工序）完成。 $Leader_k$ 负责订单的分解。

2.2 调度问题模型

根据上述定义，分布式制造调度问题可以描述为：由 $Leader_k$ 发起，按照预期要求的调度目标，将 n 个分解后的作业任务 Job_j 分配到若干制造单元 $Cell_i$ 中，并由 $Cell_i$ 完成其内部的“任务—机器”调度。

每一作业 Job_j 由多道工序组成，其技术约束条件（工艺规程、要求等）以及每个作业在每台机器上的加工时间、成本等已知，并可通过链接相关数据库查询、导入。

这里 Job_j 的调度除需满足其技术约束外，同时假定：

- 1) 每个 $Cell_i$ 可以拥有若干机器，各个机器处理的加工类型既可相同也可以不同，但这些机器设备类别、数量、加工能力一定，某种工件任务 Job_j 在各个机器上的加工时间预先确定，且在整个加工过程中保持不变；
- 2) 同一 Job_j 不同工序的加工在完成，在 $Cell_i$ 内 Job_j 的排产类似于传统车间调度中的 $n/m/G/C_{max}$ 问题；
- 3) 在系统运行过程中， Job_j 可以动态加入、也可根据全局任务的需要进行重新分配、重新调度。

3. 分布调度机制

在分布式制造系统中，每个制造单元既相互独立，又彼此联系。因此，其调度问题比传统单车间/工厂调度问题复杂得多。

3.1 分布调度问题的特点

当 Job_j 进入系统后，要选择合适的制造单元 $Cell_i$ 及机器 $Machine_l$ 来完成其加工任务，调度系统的任务包括：

- 1) 面向制造单元的制造任务优化调度，即盟主 $Leader_k$ 选择合适的制造单元 $Cell_i$ 来完成 Job_j 的加工；
- 2) 面向工件制造过程的作业优化调度，即由制造单元 $Cell_i$ 将所有承担的作业任务 Job_j 安排给合适的机器 $Machine_l$ 来完成；
- 3) 面向动态事件处理的协调调度，即在工件加工过程

中, 根据全局任务的需要, 通过各制造单元、机器设备之间的协调, 适时进行任务重新分配、重新调度, 并随时处理非预见性的急件加工、故障处理等动态事件。

上述各任务子项并非各自独立, 而是彼此关联的。例如, 任务在制造单元中分配的过程中, 各制造单元 $Cell_i$ 首先在内部“工件—机器”优化调度基础上, 提交几种备选方案, 盟主 $Leader_k$ 再从这些可行解中作出选择与决策; 另外, 工件排产过程面向制造单元内部工件的制造过程, 类似于传统车间调度, 但是在调度过程中, 不是仅依据制造单元自身的情况完成调度决策, 而是以实现全局性能前提下的个体性能优化为目标, 通过与其它制造单元的协商、协调, 完成制造单元内的作业任务优化调度。因此, 分布式调度是一个多对象、多任务、多目标复杂调度问题, 需要通过各制造单元的自主决策和相互协调来解决。本文重点研究其中制造任务优化分配过程的动态调度问题, 设计了基于遗传算法的迭代调度算法。

3.2 迭代调度算法

针对制造任务优化分配过程的协调调度问题, 设计迭代调度算法如下:

Step 1 由 $Leader_k$ 首先进行粗调度计划, 将 Job_j (包括其工艺要求、交货期等) 向各 $Cell_i$ 发布。

Step 2 $Cell_i$ 结合现有任务情况进行离线作业调度和成本计算, 提出几种预期调度方案 (包括完成时间、成本)。

Step 3 $Leader_k$ 汇总各类方案, 进行遗传优化, 寻优满足交货期、成本最低等性能指标的调度方案。

Step 4 如果得到满足的方案, 则选定该方案。

如果不能得到满足产品交货期的调度方案, 则提出成本追加等性能指标松弛方案, 提请各 $Cell_i$ 再次提交满足要求的预调度方案。转到 *Step 2*。

Step 5: 将选定调度方案通知相关制造单元, 双方核准作业任务。调度计划完成。

上述算法的实现, 通过各制造单元的并行计算完成, 单元之间信息的交互与协调通过网络来实现, 调度最佳方案的选择由遗传算法来支持。

4. 调度遗传算法

本文应用遗传算法实现分布式制造系统协调调度。遗传算法主要通过复制、交叉、变异等操作重构染色体, 从而逐渐逼近最优解。然而, 尽管遗传算法具有概率 1 收敛的极限性质, 实际算法通常难以实现理论上的收敛条件, 致使早熟收敛或收敛缓慢等问题常常出现^[5]。在分布式制造系统协调调度中, 不同制造单元、机器的组合及其结果

更复杂, 如何充分表征整个解空间的特征、保持个体多样性以及对优良基因的有效继承, 从而在全局收敛性和分散多样性之间取得平衡是成功应用遗传算法的关键。本文给出了一种改进遗传算法, 其设计原则是尽可能地保持个体多样性, 同时使子代尽量吸收父代个体的优良特征, 从而提高算法的性能。

4.1 编码方案

采用基于工序的编码方法, 它具有解码和置换染色体后总能得到可行解的优点。这种编码方法中每个工件的工序都用相应的工件号表示, 并根据在染色体中出现的次序加以编译。对于一个 n 个工件在 m 台机器上加工的调度问题, 其染色体由 $n \times m$ 个基因组成, 每个工件号能在染色体中出现 m 次, 从左到右扫描染色体, 对于第 k 次出现的工件号, 表示 Job_j 的第 k 道工序。

系统层调度仍然采用这种编码方法, 但同时将制造单元信息也赋给染色体^[1]。编码过程分为三步:

Step 1 制造单元信息编码: 对备选制造单元进行随机编码, 即将 0 到 9 或其他简单数字作为该单元的 ID 号。

Step 2 作业信息编码: 为作业任务赋予不同的 ID 号, 例如, p01 代表 01 号工件任务。

Step 3 调度任务编码: 根据作业任务的工艺文件和各制造单元提供的候选机器列表, 随机地将上述两类编码, 即工件任务的作业信息编码 (工件任务 ID 号) 与候选制造单元信息编码 (相关制造单元的 ID 号) 进行合成, 作为遗传操作的调度任务编码。

这样, 所产生的染色体的基因同时包含着制造单元及其加工机器两方面的信息。此时每个基因由两部分数字表示, 每个基因中包含的信息是作业的完成时间和成本。

例如, 某个染色体为:

[1j02-2j01-2j01-1j03-2j04-1j02-1j03-1j02-2j04]

它表示该调度任务涉及 2 个制造单元和 4 个工件作业。基因的第一部分数字 (1 或 2) 为制造单元标识, 第二部分数字 ($j0^*$) 为作业任务的标识。其中, 2j01 表示 01 号作业任务可以在 $Cell_2$ 加工, 该加工经 2 道工序完成。该染色体显示出工件作业 Job_1 , Job_2 , Job_3 和 Job_4 将分别在 $Cell_2$ 、 $Cell_1$ 、 $Cell_2$ 完成加工; 它们分别有 2 道、3 道、2 道和 2 道加工工序。

4.2 种群初始化及其选择

考虑搜索效率和质量, 基于调度规则 (如 EDD, SPT 等) 产生一个初始个体, 其余个体则在解空间均匀选取, 以使初始种群尽量分散地分布在解空间。选择操作采取保优策略, 把目前解群中最好的解直接放入下一代种群中,

同时将满足交货期的解直接进入下一代；其他基因的选择采取锦标赛选择方式。

4.3 交叉

交叉操作通过保持部分基因前后关系的“部分基因串交叉”算子实现，以便子串能够继承父串的优良特征。首先随机选取两个交叉点，交换父代个体交叉点之间的基因串，如果交叉后出现非法染色体，将根据其父代染色体的特征进行修补。例如，两个父代个体为：

$$P1 = [1p02-2p01-2p01-1p03-2p04-1p02-1p03-1p02-2p04]$$

$$P2 = [2p01-1p02-2p04-1p02-1p03-2p01-2p04-1p03-1p02]$$

部分基因串交叉后产生两个子代个体：

$$C1' = [1p02-2p01-2p04-1p02-1p03-1p03-1p02-2p04]$$

$$C2' = [2p01-1p02-2p01-1p03-2p04-1p02-2p01-2p04-1p03-1p02]$$

修补之后的子代个体为：

$$C1 = [1p02-2p01-2p01-2p04-1p02-1p03-1p03-1p02-2p04]$$

$$C2 = [2p01-1p02-2p01-1p03-2p04-1p02-2p04-1p03-1p02]$$

4.4 变异

由于基因中同时包含着制造单元及加工该工件的机器情况这两方面的信息，因此变异需要进行两步操作^[1]。首先采取常规的互换操作，即随机交换染色体中两不同基因串的位置；然后再随机地选择某个工件，从可选的制造单元列表中选择一个新的制造单元，以其 ID 号替换基因串中原有的制造单元 ID 号，并将这一操作作用于目前所有的染色体。例如，某个染色体：

$$[1p02-2p01-2p01-2p04-1p02-1p03-1p03-1p02-2p04]$$

通过第一步变异操作后变为：

$$[1p02-1p03-2p04-1p02-2p01-2p01-1p03-1p02-2p04]$$

通过第二步变异操作后进一步变为：

$$[1p02-1p03-1p04-1p02-2p01-2p01-1p03-1p02-1p04]$$

4.5 适配值函数

设计了两个适配值函数，一个与满足交货期有关，表为 F_{due} ，另一个与最小代价有关表为 F_{cost} ，分别定义如下。

$$F_{due} = \frac{T_d}{\sum_{i=1}^n \max[CT_m(i)]} \quad (1)$$

$$F_{cost} = \frac{1}{\sum_{i=1}^n CP_m(i)} \quad (2)$$

其中， CT_m 和 CP_m 分别是调度任务 Job_j 在单元 $Cell_i$ 中加

工所需的完工时间和成本， T_d 是工件任务的交货期， n 为系统中 $Cell_i$ 的数量。算法运行中，满足 $F_{due} \geq 1$ 的解会被保存起来，直接进入下一代种群，调度算法的目标是获得满足交货期而 F_{cost} 最大，即制造成本最低的调度结果。

4. 仿真实验

本文提出的方法旨在解决分布式制造环境下协调调度问题，但它同样适于传统单车间调度。为验证所提出方法的有效性，本文分别针对常规 Jobshop 调度问题和分布式调度进行了计算实验。

4.1 传统优化调度问题算例

此实验的目的是为了测试所提出调度算法的性能。取制造单元数量为常数 1，分布式调度系统退化为单一制造单元（独立车间/工厂），分布式调度转变为常规 Jobshop 调度问题。

这里选择著名的 6×6 benchmark 调度算例^[6]作为实验对象，以常用的 makespan 时间作为性能指标。作业任务的每个操作在不同的机器上以不同的加工时间完成，调度任务见表 1。表中， M_{xxx} 表示执行该操作的机器标识，因为所有机器属于同一个车间，所以操作为 $M01x$ ；括号中的数字表示操作在该机器上的加工时间。仿真实验在 Matlab 环境下进行，遗传算法初始种群个体数设为 400，迭代次数取为 58，解码方式为半活动调度。图 1 是算法运行曲线，图 2 是优化调度结果的甘特图。

6×6 benchmark 的最优解为 55，求解时间因使用不同的搜索方法、不同的初始条件而不同。从图中可以看出，本文所提出算法的加工时间为 58，算法运行时间是 5 秒（windowsxp/248MB, 4-M CPU 2.20GHz），仿真结果接近最优。

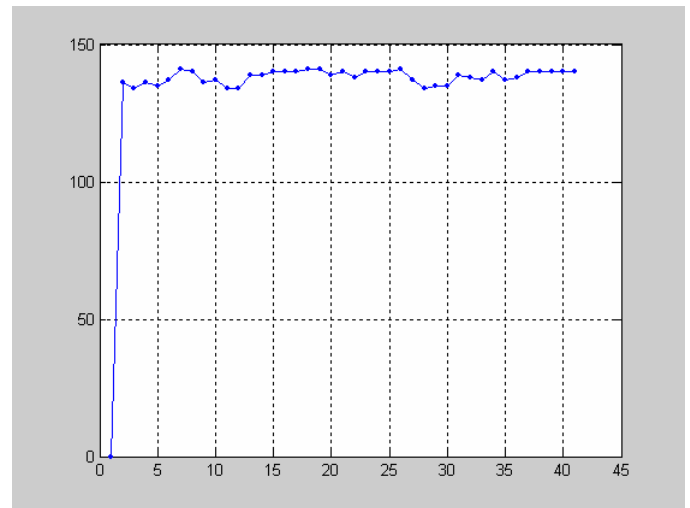


图 1 传统 6×6 调度运行结果

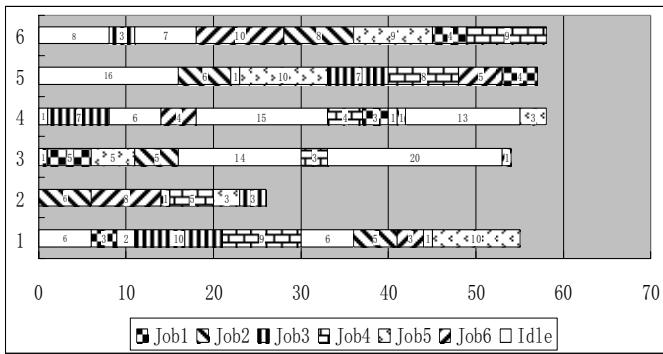


图2 传统6×6调度结果甘特图

4.2 分布式调度问题算例

此实验的目的是为了测试所提出调度算法处理分布式多制造单元动态调度问题的可行性。实验中，假设一批包括8个作业任务在3个制造单元中进行调度，每个制造单元分别有3个可用的机器，同时设定了相关机器的加工能力和加工成本。仿真结果如图3所示。由于Cell₂的设定加工成本较低，因而它只得到了2个工件的加工任务。表明调度算法能够在满足交货期的前提下，选择成本低的单元完成制造任务。

当有急件加入时，算法根据现有系统任务情况，确定急件的最低成本，并将其分配到合适的加工单元和机器。图中粗方框表示的是一个有3道加工工序的急件任务，它进入系统后，被分配到Cell₂中。由于它的加入，Cell₂中原有的调度在满足原有工件任务交货期的情况下作了一定的调整（图中如箭头所示）。

5. 结论

分布式制造是现代制造系统的发展方向，研究分布式制造调度问题具有重要的理论与应用意义。本文描述了分布式制造调度问题模型，并针对调度问题的特性，将复杂的分布调度问题，分解为面向制造单元的制造任务协同优化调度、面向工件制造过程的作业优化调度和面向动态事件处理的协调调度等调度子问题。在此基础上，提出一种基于改进遗传算法的分布调度算法，实现“制造任务——制造单元——机器设备”的协调与优化调度。仿真计算结果表明，所提出的调度策略和算法具有良好的动态性和优化能力。

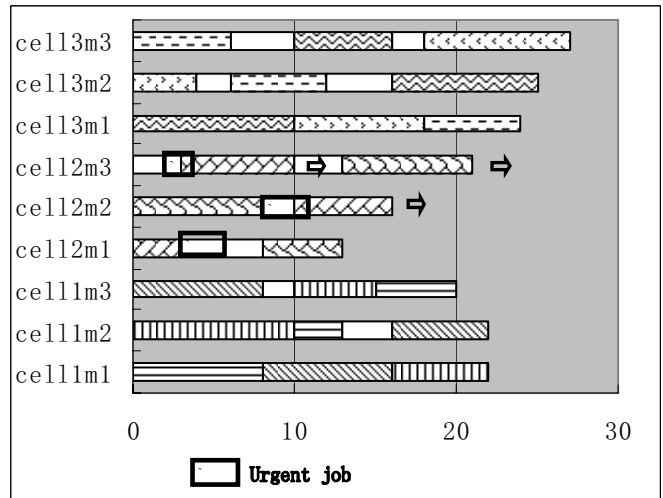


图3 分布式动态调度结果

分布式制造调度是一个复杂的问题，本文的研究只是对这一问题的初步探索，如何进一步提高遗传算法在分布制造环境下的动态处理能力、或将遗传算法的优化能力与强于分布式问题求解的其他方法有机结合是本文下一步的工作。

参考文献

- [1] H.Jia, A.nec, J.Fuh and Y.Zhang, "A modified genetic algorithm for distributed scheduling problems", *Journal of Intelligent Manufacturing*, vol. 14, pp. 351-362, 2003.
- [2] 王家海, 卢洪波, 张曙. "用于协调调度的遗传算法的实现". *机械科学与技术*, vol. 21, pp. 637-639, 2002.
- [3] Y.H. Wang, C.W. Yin and Y. Zhang, "A multi-agent and distributed ruler based approach to production scheduling of agile manufacturing systems," *International Journal of Computer Integrated Manufacturing Systems*, vol.16, pp. 81-92, 2003.
- [4] 李培根, 张洁. 敏捷化智能制造系统的重构与控制. 北京: 机械工业出版社, 2003.
- [5] 王凌. 车间调度及其遗传算法. 北京: 清华大学出版社, 2003.
- [6] J.F.Muth, and G.L.Thompson, *Industrial scheduling*. Prentice-Hall, Englewood Cliffs, 1963.