基于遗传算法的车间作业调度系统研究

纪树新　钱积新　孙优贤
(浙江大学工业控制技术研究所·杭州,310027)

摘　要　在车间作业调度数学表达模型的基础上,讨论了"基于遗传算法的车间作业调度系统"整体设计,给出了主要包括 JSS 仿生、JSS 遗传进化与 JSS 仿真三个方面在内的系统设计模型。应用实例说明了设计的可行性与成功性。

关键词　遗传算法,仿真,仿生,遗传进化,基因

1 引　言

车间作业调度(Job Shop Scheduling,简称 JSS)是一个典型的 NP 难问题,是 CIMS 领域中研究的重要课题。长期以来,JSS 研究的方法始终以启发式算法为主导,绝大部分的 JSS 研究工作也都围绕着启发式算法进行。尽管这些研究取得了一定的应用效果,但却存在着难以克服的弱点,如计算规模不可能较大、寻优结果不具备全局特性等等。近年来,又有学者提出了基于神经网络的车间作业调度系统,但此种方法在 JSS 规模较大时,存在着计算速度慢与结构参数难以确定的弱点。由此可见,要想进一步研究 JSS,选择一种有效的方法极为必要。为此,我们提出了基于遗传算法的车间作业调度研究。基于遗传算法的 JSS,虽然国内外已有研究,但这些研究都不是直接对 JSS 应用遗传算法,而是通过成组技术来对 JSS 应用遗传算法,这就使得 JSS 研究具有一定的局限性,即受到成组技术研究与其适应性的限制,而我们的研究则是将遗传算法直接应用到 JSS 问题中,消除了目前遗传算法只能应用于成组技术 JSS 的局限性。

2 车间作业调度的数学模型

我们知道,车间作业就是利用车间资源对某一对象进行生产的过程,作业调度则是对车间作业进行有效排序,使某个目标函数最小。作业的对象既可以是某个部件,也可以是某个零件;既可以是某个零件的某个工序,也可以是某个工序的某个工步。由于工序是部件和零件的基本元素,同一工序的工步通常是工序的绝对分割,即在同一台机器上连续完成,所以我们将车间作业的对象确立为零件的工序。于是,作业调度的对象也就确立为工序。

车间作业调度不仅具有极值性,而且具有有效性。车间作业调度的基本约束条件是:

1) 每道工序必须在它指定的机器上加工生产。

2) 每道工序必须在它前面的工序加工生产完毕后再加工。

3) 同一时刻同一台机器只能加工一个零件。

为了给出 JSS 的数学表达模型,我们先给出如下几个定义。

定义 1　设 P 代表 n 个工件的集合,则 P 可以表示为 $P=\{P_1,P_2,\cdots,P_n\}$。

定义 2　设 M 代表 m 台机器的集合,则 M 可以表示为 $M=\{M_1,M_2,\cdots,M_m\}$。

定义 3　设 P_i 的工序数目为 k_i,JP_i 代表 P_i 工件的工序集合,则 JP_i 可以表示为 $JP_i=$

$\{J(1),J(2),\cdots,J(i_k)\}$。

定义 4 设零件 P_i 中的工序 —J_i 中的元素在机器 M 上的加工次序为 $M_{i(1)},M_{i(2)},\cdots,M_{i(k)}$，记为 $i(1),i(2),\cdots,i(k)$，其中 $k=\max\{k_1,k_2,\cdots,k_n\}$，$0\leqslant \mathrm{val}(i(j))\leqslant m$，$0\leqslant i\leqslant n$，$0\leqslant j\leqslant k$，当 $j>k_i$ 时，$\mathrm{val}(i(j))=0$。则机器加工次序阵 Q 可以表示为

$$Q=\begin{bmatrix}1(1) & 1(2) & \cdots & 1(k)\\ 2(1) & 2(2) & \cdots & 2(k)\\ \cdots & \cdots & \cdots & \cdots\\ n(1) & n(2) & \cdots & n(k)\end{bmatrix}$$

定义 5 设机器 M_i 上的加工工序排列为 JM_i，其中 $0<i<m$，则 JM_i 可以表示为 $JM_i=\{j(1),j(2),\cdots,j(l_i)\}$，其中 l_i 为在机器 M_i 上加工工序的总数，$j(i)$ 为机器 M 的第 i 个加工的零件工序代号。

定义 6 对定义 5，如果设 $w=\max\{l_1,l_2,\cdots,l_m\}$，且当 $w>l_i$ 时，JM_i 中的元素 $j(w)$ 有 $\mathrm{val}(j(w))=0$，则可以用一个矩阵 JM 表示所有机器的工序排列，我们称 JM 为机器加工工序排列阵，记为 JM，其中，JM_i 代表第 i 台机器上的加工工序排列阵。

$$JM=\begin{bmatrix}1(1) & 1(2) & \cdots & 1(w)\\ 2(1) & 2(2) & \cdots & 2(w)\\ \cdots & \cdots & \cdots & \cdots\\ m(1) & m(2) & \cdots & m(w)\end{bmatrix}\quad 或\quad JM=\begin{bmatrix}JM_1\\ JM_2\\ \cdots\\ JM_m\end{bmatrix}$$

基于上述定义，我们可以给出 JSS 数学模型的定义：

定义 7 设 P 为工件集合，M 为机器集合，Q 为机器加工次序阵，JM 为机器加工工序排列阵，如果对于一个给定的机器工序排列阵 JM_0，满足如下关系式 $F(JM_0)=\min F(JM)$，即 JM_0 使目标函数 $F(JM)$ 取值最小，且与 Q 相容，则称 JM_0 为 JSS 的最优解，而称求解 JM_0 的过程为车间作业调度。

从定义 6 我们知道，组成 JM_0 的 JM_i 彼此是相互独立的，即任一机器加工工序排列阵中元素排列次序的变化并不会影响其它机器的机器加工工序排列阵的结果。我们称 JSS 数学模型的这种属性为 JSS 数学模型的内在独立性。

3 系统整体模型

基于遗传算法的车间作业调度系统(Job Shop Scheduling System Oriented Genetic Algrithms，简写为 JSSSOGA)可由上述的 JSS 数学模型及其 GA 的有关内容给出，即 JSSSOGA 由三个子系统构成：材料输入子系统 MI，遗传算法子系统 GA 及产品输出子系统 PO。三个子系统之间是通过数据库联系和通讯的。MI 的主要功能是完成模块 GA 所需要的“原材料”，即产品定义信息、产品制造信息、产品需求信息，等等。这些输入信息经过处理之后，将存入相应的数据库中。PO 的功能主要是根据用户所提供的输出需求模式，将优化结果转化为用户所需求的表达方式，并将其输出到输出设备上或系统成品库中。GA 的主要功能是根据 MI 模块所产生的产品定义、制造、需求数据库完成 JSS，它的结构模块图如图 1 所示。

由图 1 可知，GA 模块主要由三个处理模块构成：系统仿生模块 EB，遗传进化模块 GE 和系统仿真模块 SS。EB 模块主要完成遗传进化所需的生态环境的建立，GE 主要根据 EB 产生

的生态环境完成遗传算子操作，而 SS 则是根据 GE 需求进行 JSS 仿真。

上面，我们给出了 JSSSOGA 系统模型的概貌，由于 GA 是 JSSSOGA 系统的核心子系统，是我们研究和开发的主要内容，所以下面着重介绍 GA 设计，即 GA 的三个组成模块的详细设计。

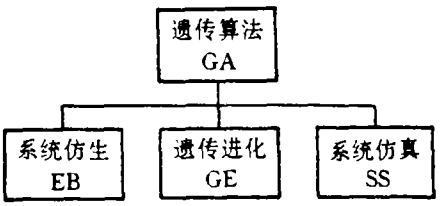

图 1 GA 结构模块图

4 JSS 遗传算法设计

4.1 系统仿生

由于遗传算法 GA 是以生物群体遗传进化理论为依托，所以使用遗传算法 GA 解决实际问题的首要工作是建立起利于研究的遗传进化的对象。那么，怎样把一个实际问题转化为遗传算法 GA 所需要的“基因”、“染色体”和“群体”，这实质上正是系统仿生处理模块(EB)所要解决的问题。EB 主要包含三个处理模块：基因培植处理模块 GB，染色体培植模块 CB 和群体培植模块 PB。

4.1.1 JSS 基因培植

基因培植处理模块 GB 主要包含两个处理模块，一个是基因原子制作处理模块 GMC；一个是基因库制作模块 GBC。GMC 模块的主要功能是对 MI 主模块所产生的产品定义数据库、产品制造数据库、产品需求数据库中的数据进行格式化，形成制作基因所需要的材料——对象编码，并将其存入相应的数据库中。GBC 模块的主要功能是根 GMR 所产生的基因材料制作系统所需的基因库。

4.1.2 JSS 染色体培植

与生物学相仿，JSS 染色体就是 JSS 基因的载体。那么 JSS 基因在 JSS 染色体上以什么方式存在，JSS 染色体在系统中又是以什么方式来表达，这实质上是一个 JSS 染色体编码问题，它与遗传算法应用过程中的编码相对应，同时，它也是 JSS 染色体培植模块的功能。在该模块中，我们提出了 JSS 连锁基因编码法，关于它的详细论述请参见文献[1]。根据 JSS 染色体连锁基因编码法，我们就可以利用 JSS 基因创建 JSS 染色体，其过程完全通过 C++语言实现。

4.1.3 JSS 个体培植

一个 JSS 个体就是一个有效的 JSS 染色体编码字串，而 JSS 群体就是一个有限数目的 JSS 个体的集合。既然 JSS 群体是一个有限数目的 JSS 个体的集合，则 JSS 群体培植问题也就转化为如何根据 JSS 染色体编码字符串生成有限数目的 JSS 个体问题。JSS 群体培植过程中需要做到如下几点：一是产生的 JSS 个体应具有随机性。这一点可以通过对 JSS 染色体编码串随机变换得到保证，从而保证群体具有良好的分散性；二是产生的 JSS 个体不应重复。这一点可以通过与存在 JSS 个体进行比较检验得到保证。

按照上述，运用 C++语言，通过对 JSS 染色体的操作运算，我们便可以培植出系统所需的 JSS 群体。

4.2 JSS 遗传进化

遗传进化 GE 模块以 Holland 的遗传算法和 JSS 数学表达模型为基础设计三个子块：参数配置模块 PS，算子运作模块 OO 和控制反馈模块 CF。

4.2.1 遗传进化参数配置

遗传进化的算子运作时，涉及到大量参数，我们将这些参数统称为遗传进化参数。遗传进化参数必须在系统遗传进化算子运作之前进行配置，担负这一功能的模块就是PS。PS模块通过与参数库交互信息，实现了在线配置与离线配置这样一个参数配置的双重功能。目前，所有GA中的遗传进化参数都是模仿自然生物的遗传进化提出来的，我们对此进行了综合并加以修正，具体见文献[1]。

4.2.2 JSS遗传进化算子运作

系统的遗传进化算子主要包括选择(Selection)、交叉(Cross)、变异(Mutation)、组合(Combination)。算子的运算与操作是JSS遗传进化算子操作模块的功能。关于JSS遗传进化算子的详细设计参见文献[1]。

4.2.3 JSS遗传进化反馈与终止

在系统仿生条件一定的前提下，影响系统遗传进化的主要因素是遗传进化参数和算子。由于遗传算子的类型包含于遗传进化参数中，所以，可以说系统仿生条件一定前提下的影响遗传进化的因素是遗传进化参数。由于遗传进化参数是用户预先通过与系统交互自由设定的，这就使得遗传进化参数的设定具有一定的主观性与偶然性。如何对系统遗传进化参数进行一定程度的调整，进而指导遗传进化向着有力的方向发展，这正是遗传进化反馈模块的职能。

CF模块主要是由两个模块组成，一是信息采集模块IP；一是信息反馈模块IF。IP模块的功能是对遗传进化过程中的主要信息进行采集，其中，包括遗传计划产生的结果信息和使用的参数信息，然后将其存入相应的数据库中，供IF模块使用；IF模块的功能是分析IP模块采集的信息，并将信息反馈给系统和用户，指导系统和用户修正遗传进化参数，控制遗传进化的发展，它包括手工控制和自动控制两种形式，其中自动控制使用了遗传进化参数精简GA优化法，具体参见文献[1]。

根据JSS的实际需求，我们确立了两个JSS遗传进化终止条件，即通过遗传进化参数GENERATIONS人为控制遗传进化收敛所需要的遗传进化操作次数，从而终止JSS遗传进化；通过计算JSS遗传进化最优个体适应度均方差，自动终止JSS遗传进化。系统实际运行表明，前者比后者更为有效。

4.3 JSS仿真

4.3.1 JSS个体适应度

对JSS个体Indivadual适应度，给出如下定义

$$\text{Fitness(Indivadual)} = -\sum_{i=1} k_i \text{Item}_i\text{(Indivadual)} \tag{1}$$

其中，Item_i代表JSS优化所涉及的项目，又称优化目标；k_i为对应项目的罚系数，又称目标系数，且$0 \leqslant k \leqslant 1$。

式(1)是一个通用的JSS个体适应度表达式，根据问题的实际需要，我们将式子(1)的优化目标确定为二：一是MAX，即机器完全完成时间；一是AVE，即零件平均产出时间。于是，式(1)就变成下式

$$\text{Fitness(Indivadual)} = -(K_{\max} \cdot \text{MAX(Indivadual)} + K_{ave} \cdot \text{AVE(Indivadual)} \tag{2}$$

我们在系统中把式(2)作为JSS个体的适应度定义。

4.3.2 作业调度仿真

从JSS个体适应度定义可知，要想求出JSS个体的适应度，必须首先求出MAX和AVE。

然而，由于 JSS 的复杂性，依赖于现有的数学工具不可能求解 MAX 与 AVE，为此，我们提出了运用系统仿真求解 JSS 个体适应度的思想，设计了 JSS 仿真系统。

JSS 仿真主要包含如下功能：

1）对仿真个体进行译码，将其转换为仿真所需要的 JSS 数学表达模型。

2）检验仿真个体的有效性。

3）对 JSS 个体进行实时仿真，计算机器完全完成时间与机器平均完成时间等指标。

4）根据 JSS 遗传进化要求计算 JSS 个体适应度。

我们已经利用 C++语言完全实现了 JSS 仿真系统。具体见文献[1]。

4.4 应用算例与结论

按照上述 JSS 遗传算法设计，我们开发了“基于遗传算法的车间作业调度系统”(JSSSOGA)，为了检验系统开发的成功性，我们设计了许多算例。通过对 JSSSOGA 系统进行若干算例的试验和实例的应用，都获得了令人满意的 JSS 解(限于篇幅，算例省略)。由此也充分说明了 JSS 遗传算法是解决 JSS 问题的有效方法，同时也说明了 JSS 连锁基因编码法与 JSS 遗传进化算子设计的可行性与成功性。

参 考 文 献

1 纪树新．基于遗传算法的车间作业调度系统研究．博士论文，1995

2 Goldberg D E. Genetic algorithms in search, optimization and machine learning MA: Addison－Wesley, 1989