

基于混合遗传算法的 JIT 投产顺序计划研究

蔡志峰¹, 李 萍¹, 曾忠晨¹, 张阿卜²

(1. 厦门大学自动化系, 福建 厦门 361005;

2. 厦门大学计算机与信息工程学院 CIMS 实验室, 福建 厦门 361005)

摘 要: JIT 管理方式是实施 CIMS 工程中的一种重要的生产管理方式,以消除制造过程中的一切浪费为宗旨,消除和降低企业生产经营中的各种浪费,降低产品的成本,从而给企业带来了巨大的效益。JIT 管理方式近来被广泛地应用于企业的生产经营管理中。在 JIT 生产方式中如何设计最佳的投产顺序计划问题一直是一个难题。在前人研究成果的基础上,使用把原有算法和自调整参数遗传算法相结合的混合遗传算法来求解这个问题。仿真结果表明这种算法是行之有效的。

关键词: 计算机集成制造系统; 准时生产; 混合遗传算法

中图分类号: TP391.7 **文献标识码:** A

JIT Production Sequencing Based on Blended Genetic Algorithms

CAI Zhi-feng¹, LI Ping¹, ZENG Zhong-chen¹, ZHANG A-bu²

(1. Dept. of Automation, Xiamen University, Xiamen 361005, China;

2. CIMS Lab., School of Computer and Information Engineering, Xiamen University, Xiamen 361005, China)

Abstract: JIT is an important way for a CIMS project. Aiming to eliminate all wastes in the process of manufacture, it is effective to eliminate and reduce all kinds of wastes in production management, and cut down the cost of products, thus bringing the manufacturer to great benefits. So JIT has been widely used during the past years. In JIT mode of production, it is a difficult problem on how to design the optimal sequence. In this paper, Blended genetic algorithms (BGA) combining the proximal solution with genetic algorithms (GA) is used to solve the problem. The simulations show that the BGA is effective.

Keywords: CIMS; Just in time; Blended genetic algorithm

1 引言

计算机集成制造系统 CIMS 是一项高新技术,其特征是能指导企业以全局的系统的观点来规划营造和调整经营模式,采用信息技术、自动化技术和先进的管理技术,完成产品的设计生产管理经营决策和市场服务,以实现企业产品的 TQCSE 目标^[1]。准时生产(just in time, JIT)管理方式是实施 CIMS 工程中的一种重要的生产管理方式。JIT 是以消除制造过程中的一切浪费为宗旨,将制造过程中一切不能增加产品附加价值的因素都视为浪费,并采取各种方法和措施,消除和减少企业生产经营中的各种浪费,降低产品的成本,从而给企业带来巨大的效益。近年来这种管理方式被广泛应用于企业的生产经营管理中。然而,关于 JIT 生产方式中产品投产顺序计划的制定问题一直是一个非常困难的问题^[2]。本文在参考以往文献资料的基础上,结合遗传算法的理论提

出了一种求解产品最佳投产顺序的混合遗传算法,仿真结果表明该算法是行之有效的。

2 基于 JIT 思想的投产顺序问题的数学模型及其近似解法

2.1 数学模型

以看板为主要管理工具的 JIT 生产方式,是一种主导型的管理方式。真正作为生产指令的产品投产顺序计划只下达到最后一道工序,而下达给最后一道工序以外的计划只是每月大致的生产品种和数量计划。可见下达给最后一道工序的生产质量指挥着整个企业生产过程,其合理与否关系重大,故而必须有正确的产品投产顺序计划,这是实现适时生产的关键。制定正确的产品顺序计划,即要使各工序的作业速度大致相同,避免由于各工序作业速度不一样而引起全线停车的现象,又要使各种零部件出现的几率保持不变,避免在制品库存。在这样的约束条件下,制定最优的投产顺序计

收稿日期:2001-08-07 修订日期:2002-01-07

基金项目:厦门大学工科专项经费资助课题(3003)

作者简介:蔡志峰(1976-),男,硕士,主要研究方向为智能控制,CIMS等。

划是非常困难的,其数学模型^[2]如下。

设产品 $A_i (i = 1, 2, \dots, N)$ 的产量为 q_i , 则总产量 $Q = \sum_{i=1}^N q_i$ 。又设生产量 Q 所需要的零件 $a_j (j = 1, 2, \dots, M)$ 的数量为 n_j , 那么单位产品中 a_j 的平均需求量 m_j 为 $m_j = n_j / Q$ 。因此, 生产 k 个产品需要 a_j 的平均数为 $k \cdot m_j$ 。在顺序计划中, 将 $k \cdot m_j$ 称为零件 a_j 的 k 个目标值。假设按照既定的投产顺序, 生产前面 k 个产品所需零件 a_j 的数量为 x_{jk} , 则 x_{jk} 为零件 a_j 的第 k 个实际值。在投产顺序计划中, 要使 a_j 的出现率为恒值, 就应该使 a_j 的实际值和目标值尽可能接近。

设 G_k, X_k 分别为零件 a_j 的第 k 个目标点和实际点, 且

$$\begin{cases} G_k = (km_1, km_2, \dots, km_M) \\ X_k = (x_{1k}, x_{2k}, \dots, x_{Mk}) \end{cases} \quad (1)$$

式中 G_k ——第 j 个分量为 a_j 的第 k 个目标值; X_k ——第 j 个分量为 a_j 的第 k 个实际值。为了使所有零件的出现几率保持不变, 就应该使 X_k 尽可能接近 G_k , 就是使

$$D_k = \sqrt{\sum_{j=1}^M (km_j - x_{jk})^2} \quad 0 \quad (k = 1, 2, \dots, Q) \quad (2)$$

若定义某个投产顺序计划 O^l 的第 k 个实际点 X_k^l 为

$$X_k^l = (x_{1k}^l, x_{2k}^l, \dots, x_{Mk}^l) \quad (3)$$

且对于 O^l , 设 Q 个目标点与实际点的距离之和 (误差) 为 D^l , 则

$$D^l = \sum_{k=1}^Q D_k^l = \sum_{k=1}^Q \sqrt{\sum_{j=1}^M (km_j - x_{jk}^l)^2} \quad (4)$$

那么使 D^l 达到最小的投产顺序 O^l 就是最佳投产顺序。

2.2 一种近似解法 (目标追逐法)

求解最佳投产顺序问题, 要想获得理论上的最优解, 从数学上来加以证明是非常困难的, 数量小时可以采用穷举法, 数量较大时可采用分支定界法等, 但计算都相当的繁琐。目标追逐法^[2]的基本思想是: 在确定第 k 个投入产品时, 以已经决定好第 1 到 $k-1$ 个产品为前提, 只是使第 k 个实际点的距离最小。同理可以确定第 $k+1, k+2, \dots, Q$ 个投入产品。设装配产品 $A_i (i = 1, 2, \dots, N)$ 时所需零件 $a_j (j = 1, 2, \dots, M)$ 的数量为 b_{ij} , 算法描述如下。

```

BEGIN
    k = 1, X0 = 0 (j = 1, 2, ..., M), S1 = {1, 2, ..., N};
DK: 求 Dki* = mini Dki} = mini, Sk \sqrt{\sum_{j=1}^M (km_j - X_{j, k-1} - b_{ij})^2}, 则 Dki*
对应的产品 Ai* 作为第 k 个投入产品即 order(k) = Ai*;
    if (产品 Ai 全部排好了顺序) Sk+1 = Sk} - {i*};
else Sk+1 = Sk};
    if (Sk+1 = (空集合)) goto END;
else/ 计算 Xjk = Xj, k-1} + bj* (j = 1, 2, ..., M); k++;
goto DK; j
END
    
```

3 用遗传算法求解投产顺序计划

遗传算法 (GA)^[3] 作为建立在生物进化原理基础上的统计启发式组合优化搜索技术, 具有全局最优搜索的能力, 为解决复杂问题提供了一个很好的工具, 近年来在各个领域得到了广泛的应用。由于混合遗传算法能够把特定领域的原有算法和遗传算法有效地结合在一起, 从而在性能上超过原有算法和遗传算法^[4]; 本文依据投产顺序计划问题的特点, 结合 2.2 节给出的近似解法 (原有算法), 使用自调整参数^[6] 的混合遗传算法对其求解。

3.1 遗传算法编码

依据第 2.1 节的数学模型, 用产品 A_i 的下标 i 代表某投产顺序中的某个位置生产产品 A_i , 则一种可能的投产顺序的遗传算法的编码为一个长度为 Q 的数组, 其元素中 $i = 1, 2, \dots, N$ 的个数分别为 q_i (q_i 个 i 是在数组中任意排列的)。例如, 假设要生产 5 种产品, 需要 5 种零件 ($N = M = 5$), q_i 数组为 $\{1, 2, 3, 4, 5\}$, 则一种可能的编码为: $1, 4, 5, 2, 5, 3, 4, 5, 3, 4, 5, 2, 4, 5, 3$ 。

3.2 目标函数 u 和适应值函数 f

问题的目标是要求得某个顺序 O_g^l (g 表示当前代数, l 为第 g 代的第 l 个投产顺序) 以使得该投产顺序的误差指标 D_g^l (g 和 l 意义同上) 最小, 所以定义第 g 代群体中第 l 个体 (投产顺序) 的目标函数值 u_g^l 为

$$u_g^l = \begin{cases} \max D_g - D_g^l & D_g^l < 0.85 \max D_g \\ 0 & D_g^l \geq 0.85 \max D_g \end{cases}$$

式中 $\max D_g$ 和 D_g^l ——第 g 代所有群体 (投产顺序) 的最大误差和该代第 l 个投产顺序的误差 (可依据式 (4) 算出)。则求误差最小值的问题就转化为求目标函数的最大值问题。

为了提高遗传算法的性能, 可采用适应值函数的线性比例变换, 算法如下

$$f_g^l = \begin{cases} ((u_g^l - u_g^{\text{avg}}) / (u_g^{\text{max}} - u_g^{\text{avg}}) + 1.0) \cdot u_g^{\text{avg}} & u_g^l > u_g^{\text{avg}} \\ ((u_g^l - u_g^{\text{min}}) / (u_g^{\text{avg}} - u_g^{\text{min}})) \cdot u_g^{\text{avg}} & u_g^l < u_g^{\text{avg}} \end{cases}$$

式中 $u_g^{\text{avg}}, u_g^{\text{min}}, u_g^{\text{max}}, u_g^l$ ——第 g 代群体中的平均目标函数值, 最小目标函数值, 最大目标函数值和个体 l 的目标函数值。

3.3 遗传算子

选择 (selection) 算子 采用堵盘选择和最优选择相结合的策略。首先在每个父代群体 (假设为第 g 代) $\text{pool} [p] (p: 1 - > P, P$ 表示群体规模) 中依据堵盘选择选出用来交叉的个体, 之后进行交叉并把新产生的群体保存在缓冲数组 $\text{new. pool} [p]$ 中, 同时保留第 g 代中的最优个体于 $\text{new. pool} [0]$ 中, 再依据适应值决定 $\text{new. pool} [p]$ 是否替换 $\text{pool} [p]$, 如果 $\text{new. pool} [p]$ 的适应值高于 $\text{pool} [p]$ 则替换, 否则模仿模拟退火算法 (SA)^[5] 中的 Metropolis 接受准则, 在淘汰某个投产顺序时以 0.5 的概率保留。

杂交 (crossover) 算子 采用基于次序的杂交算子。这种

杂交算子首先在父代向量上随机选取一组位置,然后把一个父代在这组位置上的元的次序强加到另外一个父代对应的元上。在编程过程中可随机地在父代中选择用来交叉的投产顺序,而后随机地产生一个长度为 Q 的二进制串,串中为 1 的位置组合就是所选取的交叉位置。就 3.1 节中的例子来看,假设随机选择的两个父代(投产顺序)分别为

父代 1 5,1,3,5,5,3,4,2,5,5,3,2,4,4,4

父代 2 5,1,5,2,5,5,3,3,2,4,3,4,4,4,5

随机产生的一个二进制串为:11110000100100,则作用该算子之后的结果分别为

子代串 1 5,1,3,5,2,5,3,2,3,5,4,4,4,4,5

子代串 2 5,1,5,2,3,5,3,5,5,4,3,2,4,4,4

变异(mutation)算子 由于编码机制不同于传统的遗传算法编码,所以采用基于次序的变异。首先随机地选取某个个体(投产顺序数组)中的两个不同的位置,如果它们的内容相同。则再随机地选取两个不同的位置,直到它们的内容不同为止,而后交换它们的内容。还是以 3.1 节中的例子来看,假设

要变异的个体为:1,4,5,2,5,3,4,5,3,4,5,2,4,5,3

随机产生的位置为:3,6

则变异之后的个体为:1,4,3,2,5,5,4,5,3,4,5,2,4,5,3。

3.4 控制参数

取群体的规模为 40(即 $P = 40$),至于交叉概率 P_c 和变异概率 P_m 则是根据每一代群体中个体的适应值在线调整,调整的规则描述如下

$$P_c = \begin{cases} k_1(f_g^{\max} - f_g^l) / (f_g^{\max} - f_g^{\text{avg}}), & f_g^l \geq f_g^{\text{avg}} \\ k_3, & f_g^l < f_g^{\text{avg}} \end{cases}$$

$$P_m = \begin{cases} k_2(f_g^{\max} - f_g^l) / (f_g^{\max} - f_g^{\text{avg}}), & f_g^l \geq f_g^{\text{avg}} \\ k_4, & f_g^l < f_g^{\text{avg}} \end{cases}$$

式中 $0 < k_1, k_2, k_3, k_4 \leq 1.0$ ——常数; $f_g^{\max}, f_g^{\text{avg}}, f_g^l$ ——第 g 代群体的最大适应值、平均适应值和个体 l 的适应值。

4 仿真及结论

假设产品数目 M 和零件数目 N 同 3.1 节中的,即 $M = N = 5$,产品零件消耗关系矩阵为

$$b = \begin{bmatrix} 2 & 2 & 4 & 2 & 2 \\ 2 & 2 & 8 & 2 & 2 \\ 1 & 3 & 4 & 2 & 3 \\ 4 & 2 & 3 & 5 & 6 \\ 6 & 4 & 2 & 1 & 2 \end{bmatrix}$$

遗传控制参数 $k_1 = 0.85, k_2 = 0.05, k_3 = 0.9, k_4 = 0.15$ 。

(1) 取产量数组 q 为:[1,2,3,4,5],则仿真数据图形结果如表 1 所示。

(2) 如果取产量数组 q 为:[20,80,50,50,20],则仿真数

据和图形结果如表 2 所示。

表 1 的结果是在 P 500 处理机上的计算结果,而表 2 是在 P 800 处理机上的计算结果。从表 1、图 1 和图 2 可以看出,使用遗传算法或是混合遗传算法的求解结果都优于近似解法所得到的结果。就第 4 节中(1)点而言,使用遗传算法或是混合遗传算法在迭代了 500 次左右就取得了优于近似解法的结果。然而,近似解法的运行时间很少!从图 3 和图 4 可以看出,使用遗传算法在进化初期进展很快,但是迭

表 1 仿真数据图形结果 1

算法	代数	时间(s)	结果误差	图形	最佳结果
原有算法	—	0.02	37.075 6	—	1,4,5,3,5,2,4,5, 3,4,5,2,4,5,3
遗传算法	1 000	196	35.314 007	4.1	3,5,4,2,5,4,1,5, 3,4,5,2,4,5,3
混合遗传算法	1 000	183	35.314 007	4.2	3,5,4,2,5,4,3,5, 1,4,5,2,4,5,3

表 2 仿真数据图形结果 2

算法	代数	时间(s)	结果误差	图形	最佳结果
原有算法	—	0.32	703.634 827	—	参看表 3
遗传算法	8 000	745.00	782.080 627	4.3	
混合遗传算法	8 000	1 304	691.489 319	4.4	

表 3 仿真数据图形结果 2 最佳结果表

近似解法	1,3,2,4,2,5,3,2,4,2,3,4,2,1,1,4,2,3,1,2,4,5, 2,3,2,4,3,2,4,2,5,3,2,4,2,3,1,4,2,3,1,2,4,5, 2,3,2,4,3,2,4,2,5,3,2,4,2,3,1,4,2,3,1,2,4,5, 2,3,2,4,3,2,4,2,5,3,2,4,2,3,1,4,2,3,1,2,4,5, 2,3,2,4,3,2,4,2,5,3,2,4,2,3,1,4,2,3,1,2,4,5, 3,3,2,4,2,5,3,2,4,2,3,4,2,1,1,4,2,3,1,2,4,5, 2,3,2,4,3,2,4,2,5,3,2,4,2,3,1,4,2,3,1,2,4,5, 2,3,2,4,3,2,4,2,5,3,2,4,2,3,1,4,2,3,1,2,4,5, 2,3,2,4,3,2,4,2,5,3,2,4,2,3,1,4,2,3,1,2,4,5, 2,3,2,4,3,2,4,2,5,3,2,4,2,3,1,4,2,3,5,2,4,2
遗传算法	1,2,4,3,1,3,2,2,4,3,2,5,4,4,2,3,2,5,2,3,4,2, 2,4,2,5,3,2,4,3,3,2,4,3,1,2,5,2,2,4,3,2,4,3, 1,2,3,5,4,2,2,3,4,1,1,2,4,4,2,1,2,3,2,3,4,2, 5,3,4,2,2,4,3,3,2,5,2,2,1,4,3,4,2,5,2,2,3,4, 2,3,4,4,1,3,2,2,4,1,2,3,2,2,1,4,5,3,2,5,4,2, 3,4,2,5,3,2,2,2,3,4,3,2,3,4,3,2,4,5,2,2,4,1, 1,2,4,3,4,5,2,3,2,2,3,4,3,2,2,5,2,4,2,1,4,3, 1,3,2,4,4,2,5,2,2,4,3,2,1,4,2,3,3,5,2,4,2,3, 2,4,2,5,3,3,2,4,2,4,1,3,2,3,5,4,2,3,2,1,5,4, 2,2,2,3,4,1,2,3,4,3,2,4,3,2,5,2,3,4,2,4,2,1
混合遗传算法	1,3,2,4,2,5,3,2,4,2,3,4,2,1,1,4,2,3,3,2,4,5, 2,3,2,4,3,2,4,2,5,3,2,4,2,3,1,4,2,3,1,2,4,5, 2,3,2,4,3,2,4,2,5,1,2,4,2,3,1,4,2,3,3,2,4,5, 2,3,2,4,3,2,4,2,5,3,2,4,2,1,1,4,2,3,3,2,4,5, 2,3,2,4,2,3,4,2,5,1,2,4,2,3,1,4,2,3,5,2,4,2, 3,3,2,4,2,5,4,2,3,2,3,4,2,1,1,4,2,3,3,2,4,5, 2,3,2,4,3,2,4,2,5,2,1,4,2,3,1,4,2,3,3,2,4,5, 2,3,2,4,5,2,4,2,3,3,2,4,2,1,1,4,2,3,3,2,4,5, 2,3,2,4,5,2,4,2,3,3,2,4,2,3,5,4,2,3,1,2,4,1, 2,3,2,4,2,3,4,2,5,3,2,4,2,1,1,4,2,3,5,2,4,2

(下转第 93 页)

3 结 论

本文讨论一类 Flow shop 的变异问题。在这类变异问题中,给定一个截止工期,截止工期比 Flow shop 最优调度中的最大完工时间小。讨论的问题是在全部工件的完工时间不超过截止工期的限制下,极小化不可行量函数。这类问题与

线性系统状态空间同构理论中的矩阵函数因子分解有着密切联系。本文证明了这一问题等价与单机调度中极小化类似的延迟量函数,将文献[2]中的某些结论推广到 $m = 3$ 的情况。同时指出 $F2 / C_j \quad D / I_{max}$ 的某些性质在 $m = 3$ 时不再成立。

参考文献:

[1] Bart H, Kroon L G. Variants of the Two Machine Flow Shop Problem Connected with Factorization of Matrix Functions[J]. European. J. Oper. Res., 1996, 91(1) : 144 - 159.

[2] Lushchakova I N. Minimizing Functions of Infeasibilities in a Two Machine Flow Shop[J]. European. J. Oper. Res., 2000, 121(2) :380 - 393.

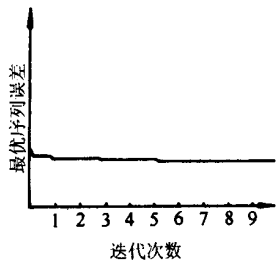
[3] Du J, Leung J Y T. Minimizing Total Tardiness on One Machine is NP-Hard[J]. Math. Oper. Res., 1990, 15(3) : 483 - 495.

[4] Fields M C, Frederickson G N. A Faster Algorithm for the Maximum Weighted Tardiness Problem[J]. Information Processing Letters, 1990, 36(1) : 39 - 44.

(上接第 52 页)

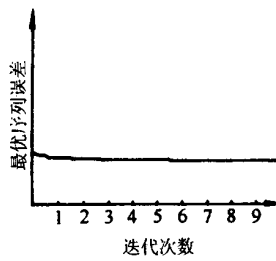
代了 8 000 次的结果不如混合遗传算法的结果。使用混合遗传算法把近似解法产生的个体(投产顺序)作为初始群体的一个(程序中存放在 pool[0]中),虽然能够在群体的交叉过程中引入好的因素,而且能够保证具有最优选择的混合遗传算法得到的解至少不会比原有算法的差,但是在迭代过程中收

敛不快(图 4 中由于初始群体中随机产生的个体的误差很大(远大于近似解法),而使得曲线很平)。不过,由于遗传算法的全局收敛性,随着迭代次数的增加,求得更好的个体(投产顺序)乃至最优的个体是显而易见的。



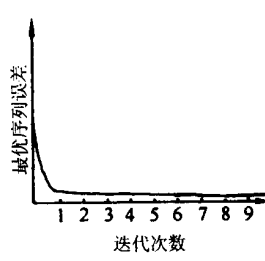
初始最大误差 :152.001 770
 初始最小误差 :52.475 521
 最终群体误差 :25.314 007
 迭代次数 :1 000
 除 :100

图 1 遗传算法计算 1 000 次的结果图形



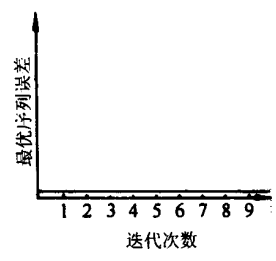
初始最大误差 :150.005 176
 初始最小误差 :43.201 100
 最终群体误差 :35.314 007
 迭代次数 :1 000
 除 :100

图 2 混合遗传算法计算 1 000 次的结果图形



初始最大误差 :20 054.980 469
 初始最小误差 :9 573.557 617
 最终群体误差 :708.006 87
 迭代次数 :8 000
 除 :800

图 3 遗传算法计算 8 000 次的结果图形



初始最大误差 :21 136.279 897
 初始最小误差 :703.634 827
 近似解误差 :703.634 827
 最终群体误差 :691.409 319
 迭代次数 :8 000
 除 :800

图 4 混合遗传算法计算 8 000 次的结果图形

参考文献:

[1] 汪定伟. 适时生产与精良生产[M]. 企业现代化新概念新技术及其应用丛书,北京:中国经济出版社,1999:79 - 93.

[2] 李芳芸. CIMS 环境下——集成化管理信息系统的分析设计与实施[M]. 北京:清华大学出版社,1996.

[3] Holland J H. Adaptation on Nature and Artificial Systems[M]. Cambridge: MIT Press, 1992.

[4] 刘勇. 非数值并行算法——遗传算法[M]. 北京:科学出版社,1995.

[5] 康立山. 非数值并行算法——模拟退火算法[M]. 北京:科学出版社,1994.

[6] 王攀. 带移民的自适应遗传算法及其应用[J]. 武汉汽车工业大学学报,1999,21(2) : 57 - 60