

化工工业多品种成批轮番生产的集成分批与调度

唐琦¹, 汪恭书², 苏丽杰²

(1. 沈阳工业大学管理学院, 沈阳 110870; 2. 东北大学信息科学与工程学院, 沈阳 110004)

摘要: 针对化工工业流程式多品种成批轮番生产集成分批与调度问题, 分析多阶段、共享设备、物料输入输出变动转化率、库存限制和品种切换调整时间的工艺特点, 建立连续时间表示的混合整数线性规划模型, 提出二维粒子群优化算法. 设计粒子编码为生产设备的加工状态, 通过有效的解码程序将粒子解释为分批和调度. 算法采用收缩算子提高局部求精能力, 并引入发散算子和速度扰动策略保持种群的多样性. 实验结果表明了所提出的算法具有良好的性能.

关键词: 化工工业; 多品种成批轮番生产; 分批与调度集成决策; 混合整数线性规划; 粒子群优化

中图分类号: C935

文献标志码: A

Integrated lotsizing and scheduling for multi-variety batch production by turns in chemical industry

TANG Qi¹, WANG Gong-shu², SU Li-jie²

(1. School of Management, Shenyang University of Technology, Shenyang 110870, China; 2. College of Information Science and Engineering, Northeastern University, Shenyang 110004, China. Correspondent: WANG Gong-shu, E-mail: wanggongshu@ise.neu.edu.cn)

Abstract: A continuous-time mixed integer linear programming model and an improved two-dimensional particle swarm optimization(PSO) algorithm are designed to tackle integrated lotsizing and scheduling for multi-variety batch production by turns in chemical industry after analyzing the characteristics of multistage, shared equipments, material input and output fluctuant conversion rate, inventory limitation and product changeover. The coding scheme of particles is designed in terms of the processing state of production units, while an effective decoding procedure translates a particle into a feasible lotsizing and scheduling solution. The improved PSO algorithm incorporates contraction operators to improve the intensification ability of the algorithm. In addition, divergence operators and velocity disturbance strategies are also introduced into the PSO algorithm to keep the diversity of the swarm. Computational results show the good performance of the proposed PSO algorithm.

Keywords: chemical industry; multi-variety batch production by turns; lotsizing and scheduling; mixed integer linear programming; particle swarm optimization

0 引言

化工工业生产的显著特点是连续性的流程式生产过程, 即物料均匀、连续不断地按一定工艺顺序流动, 在流动中物料的形态和性质不断改变, 最后形成产品. 以往, 化工工业大多采用单一品种大批量流水生产方式. 随着顾客需求日益多样化和个性化, 现代化工工业已开始转向广泛采用多品种成批轮番生产. 这种流程式的多品种成批轮番生产与以往单一品种大量生产相比, 存在着许多不同: 一是在计划期内要生产出多种产品而非一种产品, 并且多种产品通常共

用一系列生产设备; 二是多种产品要在共用生产设备上组织成批轮番生产, 每批同种产品都是连续性的流水生产, 同批内物料不可分离和间断, 这里的批是指在同一生产设备上使用同一生产配方在同一作业条件下被加工对象的集合或总量; 三是在多品种成批轮番生产中品种切换时, 不仅有生产调整准备时间损失, 而且还有物料损失, 即生产过程各阶段都存在物料转化率的问题, 输入物料并非百分之百地转化为有效输出, 物料输入输出量具有比例关系; 四是每种产品分批的批量大小受到中间库存的约束. 由于存在着

收稿日期: 2014-03-01; 修回日期: 2014-10-08.

基金项目: 国家自然科学基金项目(71202151); 教育部人文社科项目(13YJC630146).

作者简介: 唐琦(1974—), 女, 讲师, 博士, 从事生产运作管理、物流优化等研究; 汪恭书(1980—), 男, 副教授, 从事流程工业生产与物流调度、组合最优化方法等研究.

这些差异和不同,化工工业企业由单一品种大量生产转向多品种成批轮番生产时,在其生产分批、调度及其集成问题上面临着比以往更加复杂而又急需解决的强NP难问题。

迄今为止,人们对化工工业多品种成批轮番生产的计划与调度问题仍然探索较少,尽管有些文献研究了车间调度问题,但它们研究的大都是单一产品在分批次数(批次)或每批数量(批量)固定情形下,考虑两到三个生产阶段的生产调度问题。Garey等^[1]证明了生产阶段(机器设备)数大于2时的流水车间调度问题是强-NP难问题;Potts等^[2]研究了批量固定情况下问题的最优性;Chen等^[3]分析了3台设备的车间调度问题,构造了启发式求解;Baker等^[4-6]分别给出了一种产品在固定批量、变动批量情形下,两到三阶段流水车间调度建模与算法;Liu^[7]给出了启发式求解批量可变、批在流水车间各阶段加工顺序相同的问题;Cheng等^[8]研究了两阶段流水车间的最小化最大完工时间问题。针对流水车间调度问题的建模和求解方法,已有的文献主要是建立混合整数线性规划模型和构造启发式算法^[9-18]。混合整数线性规划的建模方法适用范围较广,既适用于离散时间建模,又适用于连续时间建模。Pinto等^[19]采用这种方法解决了许多石化批调度问题,而且该方法能得到调度问题的全局最优解或最优解的界,并能对解决方案做出评价。但是,这种方法对较大规模问题很难在有效时间内获得解,问题的微小变化也可能使算法失效,所以建立合理的数学模型是求解的前提和关键。启发式算法具有求解速度快的优势,该算法对简单问题可以通过分析问题的最优性来构造最优解或接近最优解,但对于规模较大的实际生产问题,解的质量很难保证。

本文结合流程式多品种成批轮番生产的实际及其计划与控制的需求,考虑到分批和调度的集成决策,以及生产工艺上的多阶段、共享生产设备、物料输入输出量间的变动转化率、品种切换调整时间和中间库存的限制,基于批之间的优先级关系,采用连续时间建模策略建立了多品种成批轮番生产分批和调度集成决策问题的混合整数规划模型,并为快速有效求解大规模问题给出了模型求解的一种改进粒子群算法。该算法采用二维编码代表批量信息以增强粒子处理约束的能力,采用前向修复策略、漂移策略和调度生成策略将粒子解码为问题的可行解,通过引入收缩和发散两方面的寻优算子以保持粒子的多样性和收敛性,旨在拓展流程式多品种成批轮番生产的分批与调度集成理论、模型及其智能求解算法,并为化工工业企业多品种成批轮番生产计划与控制提供理论依据和方法支持。

1 问题描述及建模

1.1 问题描述

给定 p 种产品 $P = \{1, 2, \dots, p\}$,每种产品的生产都要依次经过 r 个生产阶段 $R = \{1, 2, \dots, r\}$ 。产品 $i \in P$ 的需求量记为 D_i 。因为物料在各个阶段上的输入输出量具有转化率,所以产品在各个阶段上的生产量不完全相等,依赖于转化率。令 δ_{ij} 表示产品 $i \in P$ 在阶段 $j \in R$ 上的转化率。产品 $i \in P$ 在阶段 r 上的生产量为 $Q_{ir} = D_i/\delta_{ir}$,在 $r-1$ 阶段上的生产量为 $Q_{i,r-1} = D_i/(\delta_{ir}\delta_{i,r-1})$,以此类推。由于多种产品需要共用生产设备,同一种产品 $i \in P$ 在任意阶段 $j \in R$ 上的生产量 Q_{ij} 通常又都大于该阶段上设备的单次生产批量,因此产品在任意阶段上都要进行分批生产。考虑到批量大小及品种切换次数对生产经济性和安全性的影响,实际生产中都规定每次加工批量的下限和上限,令 B_{ij}^{\max} 和 B_{ij}^{\min} 分别表示产品 i 在阶段 j 上的最大和最小生产批量。因此,多品种成批轮番生产面临的第1层决策问题就是分批,即确定产品 i 在阶段 j 上的批次和批量。

产品 $i \in P$ 的一个批次在阶段 $j \in R$ 上的生产时间包括两部分,一部分是该批的额定加工时间 α_{ij} ,另一部分是与批量大小呈线性关系的生产准备时间,记 β_{ij} 为与批量大小相关的生产准备时间系数。在同一阶段 $j \in R$ 上,如果前后两个批是分别加工不同的产品 i 和 i' ,则需要一个调整时间 $\tau_{ii'j}$ 来清理设备(如化工反应器)。对于同一种产品 i ,在其相邻两个生产阶段 j 和 $j+1$ 之间都配置有存储设备,用以临时存储半成品,起到库存缓冲的作用,记存储设备的最大库存能力为 E_{ij}^{\max} 。因此,多品种成批轮番生产面临的第2层决策问题就是批调度,即在满足设备能力、产品供料关系和中间库存能力等约束条件下,确定每一阶段上各个批的开始加工时间。

1.2 数学模型

由于所研究的问题包含分批和批调度两层决策,这里通过引入批和各阶段上加工位置之间的分配关系来确定分批,进而通过引入连续时间变量来确定批调度。受批生产能力限制,产品 i 在阶段 j 上的分批数显然在集合 $\{[Q_{ij}/B_{ij}^{\max}], [Q_{ij}/B_{ij}^{\max}] + 1, \dots, [Q_{ij}/B_{ij}^{\min}]\}$ 内。阶段 j 上最大的分批数为 $\sum_{i \in P} [Q_{ij}/B_{ij}^{\min}]$,对于所有阶段而言,最大可能的分批数 $n = \max_{i \in P, j \in R} [Q_{ij}/B_{ij}^{\min}]$ 。为了建模需要,在任意阶段 j 上,可预先设定 n 个位置,每个位置可能对应一个分批,称为实位;有些位置可能不对应任何分批,称为空位。令 $N = \{1, 2, \dots, n\}$ 为预定位置集合。

$w_{ijk} = 1$ 表示产品 $i \in P$ 的一个分批被分配到阶段 $j \in R$ 上的第 $k \in N$ 个位置上, 否则 $w_{ijk} = 0$; b_{ijk} 表示产品 $i \in P$ 的被分配到阶段 $j \in R$ 上的第 $k \in N$ 个位置上的批量; ts_{ijk} 表示产品 $i \in P$ 的被分配到阶段 $j \in R$ 上的第 $k \in N$ 个位置上的批开始加工时间; $E_{ijkk'}$ 表示产品 i 在阶段 j 的位置 k 加工完输出物料, 在阶段 $j+1$ 的位置 k' 开始加工输入物料后的中间量。

本文以最小化多品种成批轮番生产的最大完工时间 m 为目标, 建立分批和调度集成决策的混合整数规划模型如下:

$$\min m = \max\{ts_{irn} + \alpha_{ir}w_{irn} + \beta_{ir}\delta_{ir}b_{irn} | i \in P\}. \quad (1)$$

$$\text{s.t. } w_{ijk}B_{ij}^{\min} \leq b_{ijk} \leq w_{ijk}B_{ij}^{\max}, \quad \forall i \in P, j \in R, k \in N; \quad (2)$$

$$\sum_{i \in P} w_{ijk} \leq 1, \quad \forall j \in R, k \in N; \quad (3)$$

$$\sum_{i \in P} w_{ijk} \geq \sum_{i \in P} w_{ij,k+1}, \quad \forall j \in R, k \in N, k \neq n; \quad (4)$$

$$\sum_{k \in N} \delta_{ij}b_{ijk} = \sum_{k \in N} b_{i,j+1,k}, \quad \forall i \in P, j \in R, j \neq r; \quad (5)$$

$$\sum_{k \in N} \delta_{ir}b_{irk} = D_i, \quad \forall i \in P; \quad (6)$$

$$E_{ijkk'} = \sum_{0 \leq d \leq k} \delta_{ij}b_{ijd} - \sum_{0 \leq d \leq k'} b_{i,j+1,d}, \quad \forall i \in P, j \in R, j \neq r, k \in N, k' \in N; \quad (7)$$

$$(h_{ijkk'} - 1)A \leq E_{ijkk'} \leq h_{ijkk'}A + (h_{ijkk'} - 1)B, \quad \forall i \in P, j \in R, j \neq r, k \in N, k' \in N; \quad (8)$$

$$(g_{ijkk'} - 1)A \leq E_{ij}^{\max} - E_{ijkk'} \leq g_{ijkk'}A + (g_{ijkk'} - 1)B, \quad \forall i \in P, j \in R, j \neq r, k \in N, k' \in N; \quad (9)$$

$$ts_{ij,k+1} \geq ts_{ijk} + \alpha_{ij}w_{ijk} + \beta_{ij}\delta_{ij}b_{ijk}, \quad \forall i \in P, j \in R, k \in N, k \neq n; \quad (10)$$

$$ts_{ij,k+1} \geq ts_{i'jk} + \alpha_{i'j}w_{i'jk} + \beta_{i'j}\delta_{i'j}b_{i'jk} - H(1 - w_{i'jk}) + \tau_{ii'j}w_{ij,k+1}, \quad \forall i \in P, i' \in P, j \in R, k \in N; \quad (11)$$

$$ts_{i,j+1,k'} + H(1 - w_{i,j+1,k'}) + Hh_{ij,k-1,k'} + H(1 - h_{ijkk'}) \geq ts_{ijk} + \alpha_{ij}w_{ijk} + \beta_{ij}\delta_{ij}b_{ijk} - H(1 - w_{ijk}), \quad \forall i \in P, j \in R, j \neq r, k \in N, k' \in N; \quad (12)$$

$$ts_{i,j+1,k'} - H(1 - w_{i,j+1,k'}) \leq ts_{ijk} + \alpha_{ij}w_{ijk} + \beta_{ij}\delta_{ij}b_{ijk} + H(1 - w_{ijk}) +$$

$$Hg_{ijk,k'-1} + H(1 - g_{ijkk'}), \quad \forall i \in P, j \in R, j \neq r, k \in N, k' \in N. \quad (13)$$

2 模型求解的启发式粒子群算法

假设 n 维空间, 问题解的构成可以用 n 维向量表示. 在算法的每一代, 粒子 q 自身找到的当前最好位置为 $P_q = (p_{q1}, p_{q2}, \dots, p_{qn})$, 整个种群找到的当前最好位置为 $G = (g_1, g_2, \dots, g_n)$. 如果用向量 $X_q = (x_{q1}, x_{q2}, \dots, x_{qn})$ 和 $V_q = (v_{q1}, v_{q2}, \dots, v_{qn})$ 分别表示粒子 q 在算法中的位置和速度, 则下面的两个等式描述了粒子速度和位置向量的更新过程:

$$v'_{qi} = wv_{qi} + c_1r_1(p_{qi} - x_{qi}) + c_2r_2(g_i - x_{qi}), \quad (14)$$

$$x'_{qi} = x_{qi} + v'_{qi}. \quad (15)$$

2.1 粒子编码

多品种成批轮番生产的分批和调度集成问题, 需要一并确定出每种产品的批次、批量以及在各阶段(设备)上的生产排序和开始生产时间. 因此, 以 px_{qlk} 表示第 q 个粒子的批量, pX_q 表示第 q 个粒子的二维位置向量, 第一维包括产品 i 和阶段 j 两种信息, 当表示产品 i 在阶段 j 上信息时, $l = i + p(j - 1)$; 第二维表示批位置.

各阶段的一个位置只能生产一批产品, 粒子 q 在阶段 j 位置 k 上的变量集合为 $B = \{px_{qlk} | l = 1 + p(j - 1), \dots, pj\}$. 在 B 中只能有一个非 0 变量. 为满足上述条件, 本文采用如下修复策略.

Step 1: 随机生成各产品的选择概率;

Step 2: 比较概率, 记录最大概率的产品编号 g ;

Step 3: 当集合 B 中的 $l \neq g$ 时, 记 $x_{qlk} = 0$.

在变量取值方面, 还要考虑生产量与需求量的关系, 因为在生产结束时中间库存为零, 各阶段生产总量应与下阶段的输入物料量相等, 所以各阶段的批量在与转化率相乘之后的总和应等于后阶段需要的输入物料量, 而在最后阶段 r 的输出量等于相应的产成品需求量. 为满足这样的内在数量关系, 这里提出如下批量修复策略.

Step 1: 为了每个阶段 j 的每种产品 i 的生产量都符合最终的要求, 令 l 表示产品和设备两种信息, Q_l 表示 l 的加工总量, S_l 表示 l 的输出总量, D_i 表示产品 i 的需求量. 初始化以上变量, $S_l = 0$, D_i 根据实际问题给出.

Step 2: 令 $j = r$.

Step 3: 计算阶段 r 的产品 i 的输出总量

$$S_l = S_{i+p(r-1)} = \sum_{k=1,2,\dots,n} px_{q,i+p(r-1),k} \delta_{i+p(r-1)}.$$

Step 4: 比较输出总量与需求量, 根据比较结果采用两种方式修复. 当输出总量大于需求总量时, 从 n 开始减少不为 0 的批, 直到输出总量等于需求量; 当输出总量小于需求总量时, 从后向前增加为 0 的批, 增加后的批量不能超过最大批量限制, 直到输出总量与需求量相等.

Step 5: 令 $j = j - 1$, 计算阶段 j 的输出量和需求量

$$S_l = S_{i+p(j-1)} = \sum_{k=1,2,\dots,n} \text{PX}_{q,i+p(j-1),k} \delta_{i+p(j-1)},$$

$$D_l = Q_{i+pj} = \sum_{k=1,2,\dots,n} \text{PX}_{q,i+pj,k}.$$

比较输出量与需求量, 根据比较结果修复.

Step 6: 当 $j > 1$ 时, 循环 Step 5 和 Step 6, 完成所有产品在各阶段设备的批量修复; 当 $j = 1$ 时, 算法结束.

2.2 粒子维数

为了提高求解效率, 本文提出一种降低粒子维数的启发式算法. 该启发式算法由两部分组成: 计算生产流程阶段(设备)的工作负荷; 计算粒子的维数.

定理 1 在多品种成批轮番生产分批和调度集成问题的最优解中, 产品 i 在阶段(设备) j 的工作负荷满足

$$\eta_{ij} \leq \frac{\alpha_{ij} D_i}{b_{ij}^{\min} \prod_{d=j,j+1,\dots,r} \delta_{id}} + \frac{\beta_{ij} D_i}{\prod_{d=j+1,j+2,\dots,r} \delta_{id}}.$$

证明 工作负荷是指加工的总时间. 由问题得

$$\eta_{ij} = \sum_{k=1,2,\dots,n} (\alpha_{ij} + \beta_{ij} \delta_{ij} b_{ijk}) = n\alpha_{ij} + \beta_{ij} \sum_{k=1,2,\dots,n} \delta_{ij} b_{ijk}.$$

因为产品 i 在阶段(设备) j 上的输出总量等于产品 i 在 $j+1$ 阶段(设备)的输入总量, 所以有

$$\sum_{k=1,2,\dots,n} \delta_{ij} b_{ijk} = \sum_{k=1,2,\dots,n} b_{i,j+1,k} \leq \frac{D_i}{\prod_{d=j+1,j+2,\dots,r} \delta_{id}},$$

$$n \leq \frac{\sum_{k=1,2,\dots,n} b_{ijk}}{b_{ij}^{\min}} \leq \frac{D_i}{b_{ij}^{\min} \prod_{d=j,j+1,\dots,r} \delta_{id}}.$$

从而有

$$\eta_{ij} \leq \frac{\alpha_{ij} D_i}{b_{ij}^{\min} \prod_{d=j,j+1,\dots,r} \delta_{id}} + \frac{\beta_{ij} D_i}{\prod_{d=j+1,j+2,\dots,r} \delta_{id}}.$$

定理得证. \square

由定理 1 可求得生产阶段(设备)工作负荷, 其计

算过程如下.

Step 1: 初始化. 令 $j = 1, \mu_j = 0$ 表示计算从第 1 个阶段的设备开始, 各阶段的设备负荷初始值都是 0.

Step 2: 计算设备负荷. 将设备 r 上所有产品的负荷相加, 令 i 取 $1 \sim p$, 循环计算 $\mu_j = \mu_j + \eta_{ij}$.

Step 3: 设 $j = j + 1$, 如果 $j > r$, 则转到 Step 4, 否则转到 Step 2.

Step 4: 计算总负荷. 设 $\nu = 0$, 令 j 取 $1 \sim r$, 循环计算 $\nu = \nu + \mu_j$.

Step 5: 输出总负荷 $C = \nu$, 算法终止.

各阶段的任务平均批加工时间为

$$\bar{u} = \sum_{j=1,2,\dots,r} \sum_{i=1,2,\dots,p} (\alpha_{ij} + \beta_{ij} \delta_{ij} B_{ij}^{\max}) / rp.$$

粒子的维数计算为 $n = \lceil C / rp\bar{u} \rceil$.

2.3 收缩和发散算子

由于基本粒子群只是向个体和全局最好解趋近的收缩寻优, 为克服其算法存在过早收敛或使粒子快速收敛于局部最优解的缺点, 引入粒子群朝着收缩和发散两个方向寻优的算子, 以使粒子群保持多样性并具有更好的收敛率. 该算法的流程如下.

Step 1: 设定粒子的种群集合 A , 给出粒子数量 m . 确定粒子的维数 $rp \times 2n$ 以及搜索空间, 搜索空间是各维粒子的取值范围.

Step 2: 初始化粒子群, 产生各粒子的初始解以及粒子更新速度的初始值.

Step 3: 定义 pb_{qlg} 为当前粒子最好位置, o_{lg} 为所有粒子的最好位置, pv_{qlg} 为当前粒子速度. 提出速度进化方程, 以产生新的速度值. 进化方程式为

$$\text{pv}'_{qlg} = w\text{pv}_{qlg} + \psi c_1 r_1 (\text{pb}_{qlg} - \text{px}_{qlg}) + \psi c_2 r_2 (o_{lg} - \text{px}_{qlg}).$$

与基本粒子群不同, 后两项在向个体和全局最好解趋近时, 加入了扰动因子 ψ , 其值为 1 或 -1, 分别代表两个相反的搜索方向. 当 $\psi = 1$ 时, 改进的粒子群算法与基本粒子群相同; 当 $\psi = -1$ 时, 粒子要离开个体和局部最好解, 这样粒子就可以跳出局部最好解的范围, 有机会在新的空间搜索到全局最好解. ψ 的取值与种群的分散度有关. 以 dir_l 代表粒子的搜索范围, 种群分散度 S 的计算式为

$$S = \frac{\sum_{q=1,2,\dots,m} \sqrt{\sum_{l=1,2,\dots,rp} \sum_{g=1,2,\dots,n} (\text{px}_{qlg} - o_{lg})^2}}{mn \sum_{l=1,2,\dots,rp} \text{dir}_l}.$$

当 $\psi = 1$ 且 $S < \sigma$ 时, $\psi' = -1$; 当 $\psi = -1$ 且 $S > \sigma$ 时, $\psi' = 1$. 其中 $\sigma = 10^{-3}$ 表示分散度的分界点.

Step 4: 根据新的速度值更新粒子的位置值, 有

$$px'_{qlg} = px_{qlg} + pv'_{qlg}.$$

Step 5: 计算目标函数值, 并与个体和全局最好解相比较, 更新个体和全局最好解.

Step 6: 循环 Step 3 ~ Step 6, 直到满足终止准则.

2.4 粒子解码

通过粒子的位置向量编码可以确定批量和批次, 粒子解码是采用调度策略给出各批加工的开始作业时间. 这样, 粒子的位置向量可以间接地表示为生产分批与调度集成问题的可行方案. 调度策略为: 每台设备沿时间轴给出一些事件点, 事件点位置不是固定的, 事件点之间的关系根据约束条件预先给定, 将批分配到事件点上, 事件点的开始与结束时间由分配在其上的批量和相关参数决定. 根据问题的实际需要, 事件点之间有时间和数量两方面的约束. ts_{ijk} 表示事件点上批的开始时间, te_{ijk} 表示事件点上批的结束时间. 当事件点上没有批生产时, 批量是 0, 开始时间与结束时间重合. 事件点之间的时间和数量约束如下:

1) 同一设备紧连的两个事件点的批结束时间与批开始时间的差为批加工时间, 结束时间大于开始时间: $te_{ijk} \geq ts_{i'jk}$.

2) 同一设备紧连的两个事件点具有时间承接关系, 后事件点的批开始时间大于前事件点的批结束时间: $ts_{ij,k+1} \geq te_{i'jk}$.

3) 相邻两个阶段的紧连事件点具有时间承接关系, 后阶段的开始时间大于前阶段的结束时间: $ts_{i,j+1,k+1} \geq te_{ijk}$.

4) 事件点之间的产品数量需考虑中间产品库存量, 由于产品 i 在阶段 j 的事件点 $k-1$ 上的批生产结束时要释放中间产品物料, 这些中间产品物料又将在设备 $j+1$ 的事件点 k 上进行批消耗, 若以 E_{ijk} 表示事件点 $k-1$ 上批释放量与事件点 k 上批消耗之差, 则 E_{ijk} 应满足 $0 \leq E_{ijk} \leq E_{ij}^{\max}$, 其中 E_{ij}^{\max} 是产品 i 在设备 j 与 $j+1$ 之间的最大库存量.

定义 1 在事件点图中, 事件点连成一系列的线, 每条事件点连成的线为相关层. 相关层分为两种: 一种是所有阶段都有事件点在连线上的, 称为完全相关层; 另一种是存在着阶段在连线上没有事件点的, 称为不完全相关层. 不完全相关层分为前端和后端, 前端不完全相关层是指没有第 1 台设备的事件点, 后端不完全相关层是指没有最后 1 台设备的事件点.

根据上述约束和定义, 给出如下事件点驱动启发式算法.

Step 1: 定义新的粒子变量, 由粒子批量为新变量

赋值, 赋值原则遵守在非完全相关层上等于 0;

Step 2: 将完全相关层依照时间轴方向编号;

Step 3: 从前向后依次操作, 按照编号从小到大的次序选择完全相关层;

Step 4: 对于确定的完全相关层, 从 $j=1$ 到 $j=r-1$ 检查 j 与 $j+1$ 之间的中间库存量, 对于小于 0 的, 移动 $j+1$ 上的批;

Step 5: 从 $j=r-1$ 到 $j=1$ 检查中间库存量是否超过最大库存限制, 对于超过的, 移动 j 上的批;

Step 6: 循环 Step 3 ~ Step 6, 直到完成所有完全相关层的操作;

Step 7: 某一批的开始时间只与本层和上层的批有关, 从前向后计算相关层上批的开始时间.

2.5 算法实现过程

结合上述编码及解码原则和相关策略, 改进的粒子群算法完整步骤如下.

Step 1: 根据编码原则定义粒子位置变量, 给出粒子的维数以及位置和速度的范围.

Step 2: 在粒子位置和速度的范围内随机产生初始种群.

Step 3: 根据调度策略, 给出问题调度时间表.

Step 4: 计算粒子的目标函数值.

Step 5: 评价种群中的粒子, 更新粒子的个体最好记录和整个种群的全局最好记录.

Step 6: 算法的全局最好解连续 G 代相同时, 对当前的粒子种群应用速度扰动策略, 从种群中随机选择一个粒子, 对其速度向量的取值重新初始化.

Step 7: 判断停止准则是否满足, 如果满足, 则转到 Step 8; 如果不满足, 则利用收缩和发散算子更新种群中粒子的速度和位置, 转到 Step 3.

Step 8: 结束.

3 仿真实验及结果

3.1 实验参数设置

为了验证改进的粒子群算法 APSO 的适用性和有效性, 本文将该算法应用于不同规模的化工工业多品种成批轮番生产的集成分批与调度问题, 仿真实例的参数设计如下: 所有产品的总需求量 D_s 分别为 50、60、70、100、200 和 300; 产品的需求量 D_i 在 $[0, D_s]$ 内随机生成, $\sum_{i=1,2,\dots,p} D_i = D_s$; 产品的品种数 $p = 5$, 生产阶段数 $r = 3$, 最小生产批量 $B_{ij}^{\min} = 5$; 最大生产批量 B_{ij}^{\max} (简称为 b)、转化率 δ_{ij} 、额定加工时间 α_{ij} 、生产准备时间系数 β_{ij} 和清理设备时间 $\tau_{ii'j}$ 分别在 $[10, 20]$ 、 $[0.6, 0.9]$ 、 $[2, 6]$ 、 $[0.1, 0.2]$ 和 $[1, 2]$ 内随机产生. 最大库存能力 E_{ij}^{\max} (简称为 em) 与最大生产批

量 B_{ij}^{\max} 有着密切关系, 在给定最大生产批量的前提下, 比值 em/b 与相邻阶段优先级关系有关, 进而影响批之间的时间间隔, 比值 em/b 分别取值为 1.0、1.5、2.0. 对于每组参数 $(G_s, em/b)$, 随机生成测试实例.

PSO 算法采用如下参数设置: $c_1 = c_2 = 2, w = 1$; 最大迭代次数 $I^{\max} = 1000$, 最大速度 $V^{\max} = 4$; 速度扰动策略中的参数 $G = 20$; 种群规模 $Swarmsize = 500$. 针对每个测试实例, 算法执行 10 次, 取 10 次运算的平均值. 算法采用 C 语言编程实现, 运行环境为一台 PC 机, CPU 为 Pentium IV 2.50 GHz.

3.2 APSO 与其他算法性能比较

APSO 的性能从解的质量和算法的求解时间两个方面衡量. 将算法解值 s 与最好解 s^* 的相对偏差 $g = (s - s^*)/s^*$ 作为度量标准, 判断算法解质量的优劣. 将本文提出的 APSO 与 Cplex 计算软件和 GA^[20] 的计算结果进行比较, 结果如表 1 和表 2 所示. 表 1 中, 上角标 1 表示 Cplex 计算软件, 上角标 2 表示 APSO; 表 2 中, 上角标 1 表示 GA, 上角标 2 表示 APSO. t 表示平均计算时间, GA 的交叉概率 $p_c = 0.7$, 变异概率 $p_m = 0.05$, 种群规模为 100.

表 1 在小规模实例集上本文算法与其他算法性能比较

instance	s^1/h	$g^1/\%$	t^1/s	s^2/h	$g^2/\%$	t^2/s
50,1.0	32.274 1	0	0.374 7	33.030 0	2.342 1	0.463 2
50,1.5	27.286 5	0	0.532 1	27.765 2	1.754 3	0.418 9
50,2.0	35.897 6	0	0.436 9	36.251 8	0.986 5	0.362 1
60,1.0	44.897 3	0	2.567 8	45.681 3	1.746 5	0.443 5
60,1.5	48.906 0	0	1.908 7	49.345 0	0.897 6	0.387 9
60,2.0	41.074 5	0	2.437 8	41.791 9	1.746 5	0.675 1
70,1.0	53.023 1	0	10.424 6	53.841 6	1.543 7	0.812 3
70,1.5	55.784 3	0	11.970 4	57.093 1	2.346 2	0.721 3
70,2.0	56.336 9	0	15.376 5	57.379 0	1.849 7	0.537 7

表 2 在大规模实例集上本文算法与其他算法性能比较

instance	s^1/h	$g^1/\%$	t^1/s	s^2/h	$g^2/\%$	t^2/s
100,1.0	97.032 1	0	3.876 9	97.032 1	0	1.423 8
100,1.5	99.797 4	0.032 1	4.354 3	99.765 4	0	1.313 9
100,2.0	103.377 6	0.976 3	5.986 1	102.378 1	0	1.075 3
200,1.0	157.026 7	0.894 3	12.345 3	155.634 9	0	5.342 2
200,1.5	167.896 5	0	14.441 9	167.896 5	0	4.141 5
200,2.0	144.032 9	0.341 2	16.896 2	143.543 1	0	6.132 9
300,1.0	245.376 6	0	30.875 2	245.376 6	0	10.378 6
300,1.5	255.681 2	0.671 1	34.769 1	253.976 8	0	14.876 0
300,2.0	237.057 7	0.936 5	45.873 4	234.858 3	0	12.843 1

从表 1 可以看出: APSO 与 Cplex 软件得到的最优解比较, 最小偏差小于 3%; 在求解时间方面, 随着问题规模的增加, APSO 的优势越加明显, 平均时间大约是 Cplex 的 10%. 表 2 的测试结果也表明, APSO 在解的质量和计算时间两个方面均优于 GA.

3.3 APSO 算法分析

为了分析收缩发散算子和速度扰动策略对算法性能的影响, 以参数组 $(G_s = 70, em/b = 2.0)$ 实例为例, 进行算法性能测试. 图 1 中给出了标准 PSO(BPSO)、带收缩发散算子的 PSO(APSO-1)、带速度扰动策略的 PSO(APSO-2) 和 APSO 四种算法的目标函数 fitness 变化曲线.

从图 1 可以看出: 在种群进化早期, APSO-1 的收敛速度快于 BPSO, 引入的收缩发散算子扩展了 APSO-1 中粒子的搜索空间, 使 APSO-1 在进化后期跳出局部最优, 并最终收敛到全局最优; BPSO 缺乏局部最优的逃逸机制, 尤其是在进化后期更容

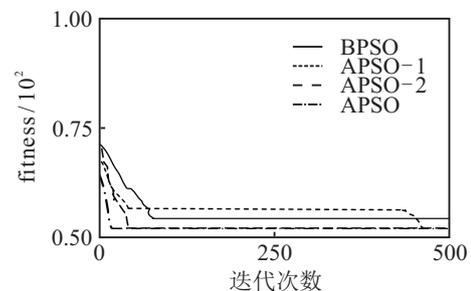


图 1 目标函数值变化曲线

易陷入局部最优位置, 因此未能搜索到全局最优; APSO-2 的性能优于 APSO-1, 说明速度扰动策略通过增加种群的多样性在一定程度上提高了算法的搜索能力; 虽然 APSO-1 和 APSO-2 都收敛到全局最优, 但是从收敛速度来看, APSO 快于 APSO-1 和 APSO-2. 综合上述实验结果, 表明引入的收缩发散算子和速度扰动策略提高了算法的全局搜索能力, 较好地避免了种群陷入局部最优值.

4 结 论

当今的化工、石油、冶金、造纸、制糖、医药制剂等流程工业企业已由过去的单一品种大批量生产转向多品种成批轮番生产。本文针对共享设备资源的流程式多品种成批轮番生产的分批与调度集成问题, 构建了基于批次之间优先级关系的混合整数规划模型, 给出了模型求解的二维编码改进粒子群算法, 拓展了流程式多品种成批轮番生产的分批与调度理论模型及其智能算法, 例证分析也弥补了至今尚无有关化工聚酯多品种轮番生产集成分批与调度研究的不足, 并且通过实例验证了所构建的模型和给出的改进粒子群算法能够克服目前已有算法只适用于小规模或模拟生产需要而解决不了大规模实际生产调度问题的缺陷, 从而为面向流程工业的实际大规模多品种成批轮番生产计划调度提供了具有应用价值的优化方法工具, 也为相关软件开发及生产实时调度与控制提供了应用参考。

参考文献(References)

- [1] Garey M, Johnson D. Computers and intractability: A guide to the theory of NP-completeness[M]. San Francisco: Freeman, 1979: 12-28.
- [2] Potts C, Baker K. Flow shop scheduling with lot streaming[J]. Operations Research Letters, 1989, 8: 297-303.
- [3] Chen J, Steiner G. Lot streaming with detached setups in three-machine flow shops[J]. European J of Operational Research, 1996, 96(3): 591-611.
- [4] Baker K, Pyke D. Solution procedures for the lot-streaming problem[J]. Decision Sciences, 1990, 21(3): 475-491.
- [5] Glass C, Potts C. Structural properties of lot streaming in a flow shop[J]. Mathematics of Operations Research, 1998, 23(3): 624-639.
- [6] Trietsch D, Baker K. Basic techniques for lot streaming[J]. Operations Research, 1993, 41(6): 1065-1076.
- [7] Liu S. A heuristic method for discrete lot-streaming with variable sublots in a flow-shop[J]. Int J of Advanced Manufacturing Technology, 2003, 22(9): 662-668.
- [8] Cheng T, Chen Z, Kovalyov M, et al. Parallel-machine batching and scheduling to minimize total completion time[J]. IIE Transactions, 1996, 28(5): 953-956.
- [9] 李光华, 陈昌领, 邵惠鹤. 批处理过程优化调度研究综述[J]. 流程工业自动化及仪表, 2002, 29(5): 1-6.
(Li G H, Chen C L, Shao H H. The survey of optimal scheduling of batch process[J]. Automation and Instrumentation in Process Industry, 2002, 29(5): 1-6.)
- [10] Shi B, Yan L, Wu W. Rule-based scheduling of single-stage multiproduct batch plants with parallel units[J]. Industrial and Engineering Chemistry Research, 2012, 51(4): 8535-8549.
- [11] Behnamian J, Ghomi S, Jolai F, et al. Realistic two-stage flowshop batch scheduling problems with transportation capacity and times[J]. Applied Mathematical Modelling, 2012, 36(2): 723-735.
- [12] 苏生, 战德臣, 徐晓飞. 一种并行机多产品厂短期调度连续时间模型[J]. 计算机集成制造系统, 2006, 12(4): 494-539.
(Su S, Zhan D C, Xu X F. A short-term continuous-time scheduling model for multi-product factory with parallel machine[J]. Computer Integrated Manufacturing Systems, 2006, 12(4): 494-539.)
- [13] 潘明, 钱宇, 李秀喜. 一种改进的顺序型多目的间歇工厂生产调度的MILP模型[J]. 流程工业学报, 2006, 57(4): 861-866.
(Pan M, Qian Y, Li X X. An improved MILP model for sequential multipurpose scheduling in the intermittent production[J]. J of Process Industry, 2006, 57(4): 861-866.)
- [14] 贺益君, 俞欢军, 成飏, 等. 多目标粒子群算法用于补料分批生化反应器动态多目标优化[J]. 化工学报, 2007, 58(5): 1262-1270.
(He Y J, Yu H J, Cheng B, et al. Multi-objective particle swarm optimization algorithm for dynamic multi-objective optimization of fill material batch biochemical reactors[J]. J of Chemical Industry, 2007, 58(5): 1262-1270.)
- [15] Mathirajan M, Bhargav V, Ramachandran V. Minimizing total weighted tardiness on a batch-processing machine with non-agreeable release times and due dates[J]. Int J of Advanced Manufacturing Technology, 2010, 48(9/10/11/12): 1133-1148.
- [16] He Y, Chen D. Immune mechanism based multi-objective ant colony algorithm approach to batch reactor Constrained dynamic multi-objective optimization problems[J]. J of Chemical Engineering of Chinese Universities, 2009, 23(4): 326-332.
- [17] Yang X, Xu Q, Li K. Flare Minimization Strategy for Ethylene Plants[J]. Chemical Engineering and Technology, 2010, 33(7): 1059-1065.
- [18] Wang I, Yang T, Chang Y. Scheduling two-stage hybrid flow shops with parallel batch, release time, and machine eligibility constraints[J]. J of Intelligent Manufacturing, 2012, 23(6): 2271-2280.
- [19] Pinto J, Grossmann I. Optimal cyclic scheduling of multistage continuous multi-product plants[J]. Computers and Chemical Engineering, 1994, 18(2): 797-816.
- [20] Mohammadi M, Fatemi Ghomi S. Genetic algorithm-based heuristic for capacitated lotsizing problem in flow shops with sequence-dependent setups[J]. Expert Systems with Applications, 2011, 38(6): 7201-7207.

(责任编辑: 齐 霖)