

文章编号: 1001-0920(2011)03-0351-06

带批量分割的多级批处理调度自组织优化

梁涛, 李歧强, 丁然

(山东大学控制科学与工程学院, 济南 250061)

摘要: 针对一类带批量分割的多级批处理调度典型问题的特点, 提出一种自下而上的自组织优化方法. 模拟人类群体“业务办理”机制, 构建了带批量分割的批处理过程调度的自组织优化模型, 分析了由批次转换和批量分割引起的复杂性, 提出了基于友好度的自组织选择策略和基于最小响应的批量分割策略, 在此基础上, 给出了自组织调度优化算法. 该方法能够在短时间内获得问题的最优解或近优解, 并通过调度实例求解结果验证了该方法的有效性和优越性.

关键词: 批量分割; 批处理; 调度; 自组织; 优化

中图分类号: TP391.9

文献标识码: A

Self-organizing optimization for scheduling of multistage batch plants with batch splitting

LIANG Tao, LI Qi-qiang, DING Ran

(School of Control Science and Engineering, Shandong University, Ji'nan 250061, China. Correspondent: LIANG Tao, E-mail: liangtao@mail.sdu.edu.cn)

Abstract: A bottom-up self-organizing approach is presented to optimize a kind of batch scheduling problems with batch splitting in multistage production processes. Firstly, inspired by people's business-processing activities, a self-organizing optimization model is built up for the scheduling of multistage batch processes with batch splitting. Secondly, the complexity resulted from sequence-dependent changeovers and batch splitting is analyzed. Then, a friendship-based selection strategy and a batch splitting strategy based on least response are introduced in detail. Based on the strategies, a self-organizing scheduling optimization algorithm is proposed for the given model. The presented approach can obtain optimal solutions or near-optimal solutions in a short time when solving problems. Finally, numerical examples show the effectiveness and the superiority of this approach.

Key words: batch splitting; batch process; scheduling; self-organization; optimization

1 引言

近 10 年来, 多级批处理过程短期调度优化问题一直备受学术界的关注, 针对该问题的研究也取得了很大进展^[1-2]. 其中大多研究都假设批处理生产过程中不允许批量的分割, 然而带批量分割的多级批处理过程的调度问题在实际化工生产中普遍可见, 用于解决该类问题的方法却比较有限. 批量分割的加入在很大程度上增加了该类问题的复杂性. 目前, 该类问题可通过建立基于离散时间描述的状态任务网 (STN) 模型^[3-4]或者资源任务网 (RTN) 模型^[5]进行求解, 但该方法在处理与顺序相关的批次切换时存在困难; 也可通过建立基于全局时间点的连续时

间 STN 模型^[6]或 RTN 模型^[7]求解, 这种方法克服了离散时间 STN 模型存在的问题, 但由于引入了大量的全局时间点而使模型规模巨大, 不易求解; Erdirik-Dogan 等人^[8]针对此类问题提出了一种基于时间槽的 MILP 模型和两层次分解调度策略, 在一定程度上可以有效地求解此类问题, 但随着问题复杂性的增加, 该方法存在复杂性不易描述和模型难以求解等难题, 对于大规模调度问题, 往往难以在有效时间内求得满意解.

自组织理论与方法在交通、传感器网络等领域的应用为此类问题的解决提供了一条新思路. 冯蔚东等人^[9]在分析交通流系统自组织特性的基础上, 提出

收稿日期: 2010-02-08; 修回日期: 2010-05-06.

基金项目: 国家 863 计划项目 (2007AA04Z157); 山东省自然科学基金重点项目 (ZR2010FZ001).

作者简介: 梁涛 (1983—), 男, 博士生, 从事复杂系统建模与优化的研究; 李歧强 (1964—), 男, 教授, 博士生导师, 从事复杂系统建模与优化算法、节能优化技术等研究.

了一种基于自组织理论的交通流管控模式,用于解决交通流优化控制问题; Collier 等人^[10]构建了一个无线传感器自组织网络系统,用于实现无线传感器网络的低功耗优化通讯. 自组织优化方法较其他组织优化方法的优势在于其更强的驾驭复杂性的能力和更好的自动趋优能力^[11]. 本文将自组织优化的思想引入生产调度领域,提出一种适合带批量分割的多级批处理调度这类典型复杂问题的自组织优化方法,并应用实例验证了该方法的有效性和优越性.

2 问题描述

本文所研究的一类带批量分割的多级批处理调度问题给出一组并行的同型或不同型的批处理设备 $\{u_j|j=1,2,\dots,m\}$ 和 n 个产品订单 $\{o_i|i=1,2,\dots,n\}$, 每个订单对应一种最终产品 $\{p_i|i \in P\}$, 如图 1 所示. 最终产品可分为两大类: 一类 $\{p_i|i \in P_1\}$ 只需经过原料的一级处理便可获得; 而另一类 $\{p_i|i \in P_2\}$ 的生产则需经过 2 个阶段的处理过程, 还需要中间产品的存储过程. 每种最终产品可以被输送到一个最终产品罐. 由于装置拓扑结构限制, 一旦产品进入最终产品罐, 则无法再次进入批处理设备, 只有进入中间储罐的产品才能被再次加工处理. 因此, 同为中间产品的最终产品 (如图 1 中的产品 A) 需要通过批量分割来满足生产需求. 每一个批处理设备可以生产一种或多种最终产品或中间产品, 而且处理批量 Q_{ij} , 批次处理时间 BT_{ij} , 批次生产成本 C_{ij} 以及批次转换引起的设备设置时间 $\tau_{i'j}$ 和设置成本 $\zeta_{i'j}$ 给定. 已知中间储罐的容量 V_i , 各订单的交货期 H_i 和各种产品的单价 CP_i 以及单位存储成本 CI_i . 调度问题的目标就是在满足生产约束和订单需求的条件下, 找出一个最优生产方案, 使得生产时间最短或生产利润最大.

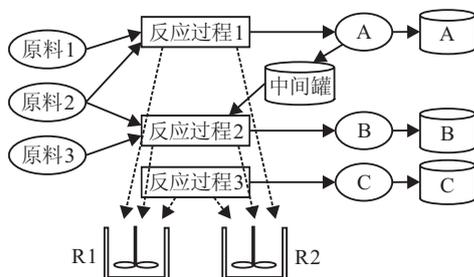


图 1 带批量分割的多级批处理过程示意图

该问题具有一定的典型性. 对于实际问题中包含多种产品或者多个交货期的订单, 都可以通过按照产品种类或交货期限的异同分别进行分割和合并, 最终转化为本文所描述的问题的形式.

3 自组织优化方法设计

3.1 自组织优化模型

自组织优化是在一定的自组织规则下, 通过系统个体的动态交互协同, 自下而上生成优化结果的一种

方式. 其中参与自组织的自主个体被称为自组织单元 (SOU). 多级批处理调度自组织优化方法的基本思想是模拟人类群体“业务办理”行为, 将整个生产环境看作业务办理大厅, 将批处理过程中的订单、批次、设备分别看作客户、客户代表和业务员, 将批次反应过程和批量分割过程看成一项项的业务, 并为每一类批量分割过程虚拟一个管理员. 这样, 批处理过程便可看作一个自组织业务办理过程.

基于业务办理机制的带批量分割的多级批处理调度自组织优化模型见图 2 (以图 1 中的问题为例). 该模型包含订单 SOU (OSOU), 批次 SOU (BSOU), 设备 SOU (ESOU), 批量分割 SOU (SSOU) 和各种存储罐. 其中每个 OSOU, BSOU 和 ESOU 分别对应实际生产中的一个订单、批次和设备. SSOU 是虚拟的批量分割决策者, 每个 SSOU 对应一种需要分割批量的中间产品, 处理其批量分割任务. 这 4 类 SOU 作为自主个体, 具有自身的属性和方法. 每个 OSOU, BSOU 或 ESOU 的属性包含了其对应实体的生产信息, 而 SSOU 的属性包含了对应产品作为中间产品和最终产品的需求和输入批量信息. OSOU 可以选择 ESOU 和创建 BSOU, 并驱使其到达相应 ESOU 接受处理, 以完成自身的产品需求; 每个 ESOU 具有一个等待队列, 等待其处理的批次位于此队列中, ESOU 可以从等待队列中选择 BSOU 进行处理, 并在处理完成后将产品按要求输出到最终产品罐或 SSOU 中; BSOU 可以根据 OSOU 的指示到达 ESOU 的等待队列; SSOU 能够按需求对输入批量进行分割. 这 4 类 SOU 的属性和方法可以方便地应用面向对象的方法来创建. 该模型中的各类 SOU 都可以根据需要方便地添加、移除或更新信息, 故该模型具有很强的建模灵活性和可扩展性.

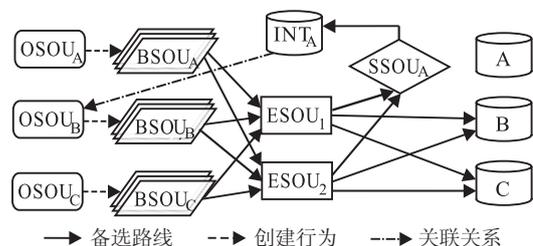


图 2 带批量分割的多级批处理调度自组织优化模型示意图

针对该自组织优化模型, 有以下几点需要说明:

1) 设中间产品集合为 P_M , 对于产品类型 $\{p_i|i \in P_1 \cap P_M\}$ 对应的 OSOU 的订单, 需求初值为完成所有产品订单对该类产品的需求量之和. 例如, 图 2 中 OSOU_A 的订单需求初值为最终产品 A 的需求量与完成产品 B 订单对 A 的需求量之和.

2) 产品 $\{p_i|i \in P_2\}$ 对应的 OSOU 与其所需中间产品的中间储罐相关联 (如图 2 所示). 只有与其关联中间储罐相连的 SSOU 中有足够多的可用中间产

品时,此类 OSOU 才会被激活去选择合适的 ESOU 和创建 BSOU.

3) OSOU 应首先选择合适的 ESOU, 然后再根据 ESOU 的设备容量创建相应批量的 BSOU. 对于批处理设备为并行同型设备(具有相同容量)时,也可以根据固定的设备容量先创建 BSOU, 再选择 ESOU 进行生产. 该情况下,两种方式效果基本相同.

3.2 基于友好度的自组织选择策略

批处理调度自组织优化模型解的产生是通过各个 SOU 的行为交互实现的, SOU 的行为准则和行为模式是决定自组织效果的关键,也是自组织优化方法的重要研究内容.

OSOU 从能够处理其批次的候选设备中选择哪个 ESOU, 主要与设备的状态、设备对该类产品的处理能力和处理成本有关. 设备的状态定义为

$$\mu_j(t) = \frac{X_j(t)}{1 + BT_{i_{1j}} + \sum_{k=2}^{Q_j(t)} (\tau_{i_{k-1}i_{kj}} + BT_{i_{kj}})} \quad (1)$$

其中: $X_j(t) \in \{0, 1\}$ 为 t 时刻设备 u_j 的可用性, 可用为 1, 不可用为 0; $Q_j(t) \in$ 自然数集, 是 t 时刻设备 u_j 的等待队列中批次的个数; $i_k \in \{1, 2, \dots, n\}$ 为设备 u_j 的等待队列中第 k 个批次的产品类型编号. 分母中 1 的作用是将 $\mu_j(t)$ 的值域限定在 $[0, 1]$. 为实现 OSOU 对 ESOU 的合理选择, 其对 ESOU 的偏好概率可表示为

$$p(E_{jk}) = \frac{(\mu_{jk})^\alpha (Q_{ij_k}/BT_{ij_k})^\beta (Q_{ij_k}/C_{ij_k})^\gamma}{\sum_{j' \in U_i} (\mu_{j'})^\alpha (Q_{ij'}/BT_{ij'})^\beta (Q_{ij'}/C_{ij'})^\gamma} \quad (2)$$

其中: α, β, γ 为 3 个参数, 分别反映设备的当前状态、设备对该类产品的处理能力和处理成本在设备被选中过程中的相对重要性, 它们的取值可根据经验或者通过仿真实验确定; U_i 为产品 p_i 对应 OSOU 的候选 ESOU 的编号集合; $p(E_{jk})$ 表示候选集合中第 k 个 ESOU 被选中的概率.

ESOU 从它的等待队列中选择哪个 BSOU 先处理, 主要与批次处理时间裕量有关. 设备 u_j 的等待队列中产品类型为 p_i 的批次在 t 时刻的处理时间裕量定义为

$$tr_i(t) = H_i - t - BT_{ij}, \quad (3)$$

则 ESOU 对等待队列中的 BSOU 的偏好概率表示为

$$p(B_{ik}) = \frac{1/tr_{ik}}{\sum_{k'=1}^{Y_j} (1/tr_{ik'})} \quad (4)$$

其中: Y_j 为设备 u_j 等待队列中的批次数, i_k 为设备 u_j 的等待队列中第 k 个批次的产品类型编号, $p(B_{ik})$ 为 u_j 的等待队列中第 k 个 BSOU 被选中的概率.

式(2)和(4)给出的 SOU 的偏好概率主要考虑了

设备处理能力、设备处理成本和订单交货期等因素对 SOU 行为交互的影响. 然而, 带批量分割的多级批处理问题中的批次转换引起的设备设置时间和设置成本使问题变得更加复杂, 同类产品的各个批次在多个批处理设备上的分散和在同一设备上处理顺序队列中的分散容易造成额外的设备设置时间和设置成本, 造成总处理时间的增加和利润的降低. 然而, 同类产品的各个批次的绝对集中会造成生产不能按时完成. 针对这一矛盾, 本文构建了基于友好度的选择策略. 为方便表述, 首先给出友好度的定义.

定义 1 假设 B_1 和 B_2 为任意 2 个 BSOU, 其对应的产品类型分别为 p_i 和 $p_{i'}$, 则称

$$\text{friend}_j(B_1, B_2) = e^{-(\tau_{ii'j})^\varepsilon \cdot \zeta_{ii'j}} \quad (5)$$

为在设备 u_j 对应的 ESOU 上 B_1 对 B_2 的友好度. 其中: $\text{friend}_j(B_1, B_2) \in [0, 1]$ 是批处理设备上批次转换引起的设备设置时间的 ε 次幂和设置成本的负指数函数, 随着批次转换代价的增大而减小; $\varepsilon \geq 0$ 为比重调节参数, 可根据经验和实际问题设定, 若批次转换引起的设备设置时间和设置成本成线性关系, 则可设置 $\varepsilon = 0$. 对于批次转换引起的设备设置时间和设置成本只与批次类型和顺序有关的情况, 式(5)可简化为

$$\text{friend}(B_1, B_2) = e^{-(\tau_{ii'})^\varepsilon \cdot \zeta_{ii'}} \quad (6)$$

定义 2 BSOU 对 OSOU 的友好度定义为与它对该 OSOU 所创建 BSOU 的友好度相等.

定义 3 定义 $R = \{R_1, R_2\}$ 为 OSOU 对 ESOU 的选择规则. 其中: R_1 表示 OSOU 考虑 BSOU 的友好度, 只选择等待队列中含有对它友好度为 1 的 BSOU 的 ESOU; R_2 表示 OSOU 无视 BSOU 的友好度, 按照偏好概率对 ESOU 进行选择.

定义 4 将 OSOU 或 BSOU 无视 BSOU 对它的友好度的概率称为 OSOU 或 BSOU 的孤僻概率. 孤僻概率的值可根据实际问题中批次转换对问题目标的影响选取, 也可通过实验的方法获得.

下面给出 OSOU 对 ESOU 基于友好度的选择算法:

算法 1 OSOU 对 ESOU 基于友好度的选择算法

Step 1: 取 $[0, 1]$ 中一个随机数, 若其小于该 OSOU 的孤僻概率值, 则转 Step 4.

Step 2: 搜索备选 ESOU 的等待队列, 依据规则 R_1 选择相应的 ESOU, 若结果唯一, 则转 Step 5.

Step 3: 若 Step 2 的结果为空, 则转 Step 4; 否则, 从 Step 2 的结果中, 依据选择规则 R_2 选择 1 个 ESOU, 转 Step 5.

Step 4: 依据规则 R_2 从候选 ESOU 中选择 1 个.

Step 5: 返回选中的 ESOU.

ESOU对等待队列中的BSOU的选择与OSOU对ESOU的选择规则类似. 本文给出以下定义:

定义5 定义 $R' \in \{R'_1, R'_2\}$ 为ESOU对等待队列中的BSOU的选择规则. 其中: R'_1 表示ESOU考虑BSOU间的友好度, 选择等待队列中对上一步完成的BSOU具有最大友好度的BSOU作为下一个处理对象; R'_2 表示ESOU无视BSOU间的友好度, 按其对应等待队列中BSOU的偏好概率选择下一个处理的BSOU.

如果ESOU上一步完成的BSOU无视其他BSOU对它的友好度或者ESOU选择的是它的第1个处理任务时, 它会遵循规则 R'_2 进行选择, 否则它会遵循规则 R'_1 选择, 而且还应考虑如果等待队列中出现时间裕量小于等于0的BSOU时则优先选择. 下面给出ESOU对BSOU基于友好度的选择算法:

算法2 ESOU对BSOU基于友好度选择算法

Step 1: 判断等待批次队列中是否有时间裕量小于等于0的BSOU. 若有, 则选中, 转Step 6; 若无, 则判断是否为第1次选择处理任务, 若是, 则转Step 5.

Step 2: 取 $[0, 1]$ 中1个随机数, 若其小于上一步完成的BSOU的孤僻概率值, 则转Step 5.

Step 3: 比较等待队列中的BSOU对上一步完成的BSOU的友好度, 遵循规则 R'_1 选择友好度最大的BSOU. 若结果只有1个, 则转Step 6.

Step 4: 从Step 3的结果中, 依据选择规则 R'_2 选择1个BSOU, 转Step 6.

Step 5: 依据规则 R'_2 从等待队列中选择1个BSOU.

Step 6: 返回选中的BSOU.

3.3 基于最小响应的批量分割策略

对于需要批量分割的产品类型, ESOU在完成该类型批次后将相应批量的产品输入SSOU, SSOU将对该批量按需求进行分割. 批量分割的加入使得问题解决过程中必须考虑产品质量的守恒和批量需求的响应顺序. 为保证生产的按时完成, SSOU确定各需求的响应顺序主要考虑需求的处理时间裕量.

定义6 最终产品需求的时间裕量定义为

$$\text{tr}_i^{\text{FIN}}(t) = H_i - t. \quad (7)$$

定义7 中间产品需求的时间裕量定义为其对应中间储罐关联OSOU所能创建的所有BSOU的最小处理时间裕量, 即

$$\text{tr}_i^{\text{INT}}(t) = \min_{j' \in U_{i'}} (H_{i'} - t - BT_{i'j'}). \quad (8)$$

其中: i' 为中间产品 p_i 对应的二级处理所得的产品类型编号, $U_{i'}$ 为能够处理生产产品 $p_{i'}$ 的设备集合.

SSOU对最终产品需求和中间产品需求的偏好

概率可分别表示为

$$p_i^{\text{FIN}} = \frac{1/\text{tr}_i^{\text{FIN}}}{1/\text{tr}_i^{\text{FIN}} + 1/\text{tr}_i^{\text{INT}}}, \quad (9)$$

$$p_i^{\text{INT}} = \frac{1/\text{tr}_i^{\text{INT}}}{1/\text{tr}_i^{\text{FIN}} + 1/\text{tr}_i^{\text{INT}}}. \quad (10)$$

因为中间产品需求的批量会根据关联OSOU选择ESOU的不同而变化, 为了最大限度地满足各需求, SSOU对中间产品需求的响应采用少量多次的方式, 每次满足1个后续批次所需中间产品的批量, 本文称之为最小响应. 下面给出SSOU进行批量分割的过程. 设中间产品可以分配给特定批次, 中间储罐中当前储量为 V_x , 其中未分配量为 V_r , SSOU中的剩余批量为 V_s .

算法3 SSOU基于最小响应的批量分割算法

Step 1: 判断是否满足 $V_s > 0$. 若否, 则转Step 6.

Step 2: 判断最终产品或中间产品是否有需求. 若否, 则转Step 5; 若只有一类需求, 则选中该需求; 若两类需求都有, 则根据偏好概率选择一种需求; 若中间产品需求被选中, 则转Step 3, 否则, 转Step 4.

Step 3: 让关联OSOU去选择ESOU, 根据所选ESOU的设备容量计算中间产品的需求量 V_b . 若满足 $V_s < V_b - V_r$, 则 $V_r = V_r + V_s$, $V_x = V_x + V_s$, $V_s = 0$, 转到Step 6; 否则, $V_s = V_s - (V_b - V_r)$, $V_x = V_x + V_b - V_r$, $V_r = V_b$, 同时激活关联OSOU, 让该OSOU创建1个相应的BSOU并驱使其到达所选ESOU的等待队列, 然后将 V_r 分配给该BSOU, $V_r = 0$, 更新中间产品需求, 转到Step 1.

Step 4: 取最终产品需求量 V_f . 若满足 $V_s < V_f$, 则 $V_f = V_f - V_s$, $V_s = 0$, 转到Step 6; 否则, $V_s = V_s - V_f$, $V_f = 0$, 更新最终产品需求, 转到Step 1.

Step 5: 将 V_s 随机输入一个相连罐, $V_s = 0$.

Step 6: 结束.

上述算法中还应注意中间储罐的当前储量 V_x 应不大于其容量值 VI , 因此每次将产品输入中间储罐时应判断是否满足该约束, 若不满足, 则应将超出部分输入最终产品罐. 另外, 中间储罐当前储量的更新还应在Step 3中创建的BSOU被开始处理时进行, $V_x = V_x - V'_b$, 其中 V'_b 为被处理BSOU所需该中间产品的数量.

3.4 自组织调度优化算法

由于各个SOU的自组织行为均基于个体局部信息, 单次自组织活动不可避免地存在产生劣质解的可能性. 为了保证自组织优化结果的全局优化性能, 基于自组织理论的外部选择原理, 本文采用多次自组织选优策略, 下面给出自组织调度优化算法:

算法4 自组织调度优化算法

Step 1: 初始化自组织环境中的各SOU及存储装

置的属性状态.若以最大利润为目标,则将当前目标最优值置为0;若以最小生产时间为目标,则将其置为 $+\infty$,给出自组织循环次数 N 和各个经验参数取值.

Step 2: 自组织运行求解.各ESOU的响应时间置为一个极大数 $H(H > H_i, i \in P)$.首先,所有订单 $\{o_i | i \in P_1\}$ 对应的OSOU按随机顺序根据算法1选择ESOU,创建相应的BSOU,并使其到达ESOU的等待序列,更新订单需求,被选中的ESOU更新状态,响应时间置为0.若OSOU的订单需求大于0,则重复该选择和创建行为,直到订单需求小于等于0.然后,ESOU按照响应时间顺序执行自组织行为,若存在当前处理批次,则完成当前处理的批次,并根据所得产品是否属于中间产品将产品分别输入到对应SSOU或最终产品罐.对应SSOU根据算法3对输入批量进行分割,并按批量的分配情况激活需要二级处理的订单 $\{o_i | i \in P_2\}$ 对应的OSOU,使其按照分配中间产品的数量进行相应的选择ESOU和创建BSOU等活动.ESOU根据算法2从等待队列中选择BSOU进行处理.如果选中,则将其响应时间设置为被选中BSOU的完成时间;否则,置为 H ,更新设备状态;若无当前处理批次,则直接执行选择BSOU和更新响应时间和状态.ESOU的执行顺序随响应时间的更新而更新,直到所有BSOU都已完成或者ESOU的响应时间都为 H ,转Step3.

Step 3: 若Step 2中获得的结果满足订单交货期要求且所得目标值优于当前最优目标值,则保存该结果作为当前最优调度结果;否则,保持当前最优调度结果不变.

Step 4: $N = N - 1$.如果 $N > 0$,则重新初始化各SOU及存储装置的属性状态,转Step 2.

Step 5: 输出最优调度结果,算法结束.

需要说明的是,如果经验参数设置不合适,则该算法可能出现找不到满意解的情况.此种情况下,应当适当修改经验参数取值或适当增大 N 值后重新计算.

4 实例求解与分析

带批量分割的多级批处理调度自组织优化模型框架和自组织优化算法采用C++进行编程实现,在CPU为2.66 GHz,内存为1 GB的PC机上进行求解.

为了验证自组织调度优化方法的有效性,本文对不同规模的4个实例进行了建模和求解,其中实例1~实例3取自文献[8].实例1的批处理过程包括2个同型批处理设备R1和R2,均可生产A, B, C, D, E共5种产品,其中产品B的生产需要产品A作为中间产品,调度期为1周;实例2除订货量与实例1不同外,其他参数与之相同;实例3的批处理过程包括4个同型批处理设备R1~R4,均可生产A, B, C, D, E, F共6种产品,其中产品B和E的生产分别需要产品A和F作为中间产品.为节省篇幅,实例1~实例3的生产过程示意图和生产参数参见文献[8].实例4的批处理过程基本与实例3相同,不同之处在于实例4中的4个批处理设备不是同型设备,其设备生产参数如表1所示,其他参数同实例3.实例1~实例4的调度目标均为在满足各生产约束和给定交货期约束下获得最大利润.

表1 实例4设备生产参数表

	批次批量				批次处理时间			
	R1	R2	R3	R4	R1	R2	R3	R4
A	60000	60000	100000	100000	12	12	20	20
B	80000	80000	120000	120000	8	8	12	12
C	70000	70000	100000	100000	9	9	15	15
D	100000	100000	150000	150000	18	18	30	30
E	70000	70000	120000	120000	18	18	30	30
F	60000	60000	100000	100000	15	15	25	25

应用本文提出的批处理调度自组织优化方法,各经验参数设置为 $\alpha = \beta = 1, \gamma = 2, \varepsilon = 0$.自组织循环次数针对4个实例分别设置为 $N = 1\ 000, N = 5\ 000, N = 20\ 000$ 和 $N = 20\ 000$.自组织优化的结果如表2所示.作为比较,表2同时给出了应用文献[8]提出的基于MILP模型的两层次分解调度方法、全局调度方法和文献[5]提出的基于离散时间RTN模型的调度方法对4个实例在同一配置PC机上进行优化调度的结果.其中基于离散时间描述的RTN模型的离散化时间取2 h.

从表2可以看出,对于小规模实例1,4种方法都可以找到最优解,而本文方法所用时间最少.对于具有紧约束的小规模实例2,只有本文给出的自组织优化方法和文献[8]给出的两层次分解调度方法能

表2 优化调度结果比较表

实例	自组织优化方法			两层次MILP ^[8]		全局MILP ^[8]		RTN-d=2h ^[5]	
	N	目标值/\$	时间/s	目标值/\$	时间/s	目标值/\$	时间/s	目标值/\$	时间/s
1	1000	302200	0.437	302200	0.624	302200	77.9	302200	10000 ^a
2	5000	1070660	2.862	1070660	14.2	1028280	20000 ^a	733580	20000 ^a
3	20000	1448603	14.254	1448543	464.48	1287315	30000 ^a	-	30000 ^b
4	20000	1452662	15.523	1448902	815.36	-	30000 ^b	-	30000 ^b

注: a. 经过相应的时间结束计算,将最佳可行解作为优化结果, b. 经过相应的时间结束计算,未找到可行解.

够找到最优解,而且在求解时间上本文方法小于文献[8]给出的方法.随着问题规模的增大,调度最优解越来越难被找到.对于较大规模的实例3和实例4,自组织优化方法与其他3种方法相比,都在更短的时间内找到了更优的可行解.以上实例可以验证本文方法的优越性,这些主要归功于合理的自组织策略和自组织涌现作用.另外还可以看出,自组织优化方法的计算时间随问题规模的增长速度远远小于文献[8]中给出的基于MILP模型的两层次分解调度方法,在求解大规模问题时优势愈加明显.然而,自组织策略的随机性,导致了自组织优化求解的随机性.应用自组织优化方法求解调度问题,并不是每次都可以获得最优解.为了测试自组织优化方法的求解稳定性,本文应用该方法针对4个实例各进行10次测试求解,应用10次求解结果偏离最优解的平均距离和平均求解时间作为衡量方法性能的标准,如表3所示.从总体上看,本文方法在求解实例问题时的求解稳定性是可以满足实际问题需求的,但随着问题规模的增大,求解结果偏离最优解的平均距离有增大的趋势.

表3 自组织优化方法求解稳定性测试结果表

实例	N	最优值/\$	平均距离/\$	平均时间/s	测试次数
1	1000	302.200	0	0.430	10
2	5000	1070.660	0	2.856	10
3	20000	1448.603	33.01	14.251	10
4	20000	1452.662	826.24	15.526	10

5 结论

带批量分割的多级批处理调度问题是实际化工生产中的常见问题之一.本文以该类复杂问题为例,将自组织优化的思想引入生产调度领域,提出了一种新型的批处理调度自组织优化方法.该方法充分考虑了批处理过程中的批次转换问题和批量分割问题,提出了基于友好度的批次、设备选择策略和基于最小响应的批量分割策略,在此基础上给出了自组织调度优化算法.实例计算结果验证了本文方法不但具有较快的求解速度,而且具有较好的寻优性能.此外,本文方法建模灵活、可扩展性强,在解决大规模复杂批处理调度问题领域具有较好的应用前景和推广价值.本文最后讨论了本文方法的求解稳定性,进一步提高本文方法在求解大规模复杂问题时的求解稳定性是作者进一步研究的目标.

参考文献(References)

[1] Floudas C A, Lin X. Continuous-time versus discrete-time approaches for scheduling of chemical processes: A review[J]. *Computer & Chemical Engineering*, 2004, 28(11): 2109-2129.

[2] Méndez C A, Cerdá J, Grossmann I E, et al. State-of-the-art review of optimization methods for short-term scheduling of batch processes[J]. *Computer & Chemical Engineering*, 2006, 30(6/7): 913-946.

[3] Kondili E, Pantelides C C, Sargent R W H. A general algorithm for short-term scheduling of batch operations I MILP formulation[J]. *Computer & Chemical Engineering*, 1993, 17(2): 211-227.

[4] Shah N, Pantelides C C, Sargent W H. A general algorithm for short-term scheduling of batch operations II computational issues[J]. *Computer & Chemical Engineering*, 1993, 17(2): 229-244.

[5] Pantelides C C. Unified frameworks for optimal process planning and scheduling[C]. *Proc 2nd Conf on Foundations of Computer-Aided Process Operations*. New York: Cache Publications, 1994: 253-274.

[6] Maravelias C T, Grossmann I E. New general continuous-time state-task network formulation for short-term scheduling of multipurpose batch plants[J]. *Industrial & Engineering Chemistry Research*, 2003, 42(13): 3056-3074.

[7] Castro P, Barbosa-Póvoa A P F D, Matos H. An improved RTN continuous-time formulation for the short-term scheduling of multipurpose batch plants[J]. *Industrial & Engineering Chemistry Research*, 2001, 40(9): 2059-2068.

[8] Erdirlik-Dogan M, Grossmann I E. Slot-based formulation for the short-term scheduling of multistage, multiproduct batch plants with sequence-dependent changeovers[J]. *Industrial & Engineering Chemistry Research*, 2008, 47(4): 1159-1183.

[9] 冯蔚东, 贺国光, 刘豹. 交通流自组织及其基于自组织理论的管控模式研究[J]. *系统工程理论与实践*, 1999, 19(9): 1-9.
(Feng W D, He G G, Liu B. Study on traffic-flow self-organization and its management-control model based on self-organization theory[J]. *System Engineering — Theory & Practice*, 1999, 19(9): 1-9.)

[10] Collier T C, Taylor C. Self-organization in sensor networks[J]. *J of Parallel Distributed Computing*, 2004, 64(7): 866-873.

[11] 师汉民. 从“他组织”走向自组织——关于制造哲理的沉思[J]. *中国机械工程*, 2000, 11(1/2): 80-85.
(Shi H M. From other-organization to self-organization — A meditation on the philosophy in manufacturing[J]. *China Mechanical Engineering*, 2000, 11(2): 80-85.)