

## 基于改进 PSO 的发酵过程同步串联混合建模

杨强大<sup>1</sup> 张卫军<sup>1</sup> 牛大鹏<sup>2</sup>

**摘要** 准确可靠的过程模型是实现发酵过程优化的基础和前提. 对于反应机理复杂的发酵过程, 串联混合建模是一种相对有效的建模方法, 但现有方法需要利用插值所得的数据进行中间变量黑箱模型的构建, 较大程度地影响了所建混合模型的泛化性能. 为此, 提出一种可将黑箱模型构建问题转化为动态模型参数辨识问题的同步串联混合建模方法, 从而避免了现有方法需利用插值数据来构建黑箱模型的不足; 通过引入多精英学习策略和惯性权重自适应调整策略, 构造了一种改进的粒子群优化 (Particle swarm optimization, PSO) 算法 — 自适应多精英学习 PSO (Adaptive multi-elite learning PSO, AMLPSO) 算法, 并采用该算法求取黑箱模型的参数; 借鉴均匀设计思想确定黑箱模型的结构. 利用诺西肽分批发酵过程实际生产数据进行实验研究, 结果验证了所提方法的有效性.

**关键词** 发酵过程, 同步串联混合建模, 粒子群优化, 算法改进, 均匀设计

**引用格式** 杨强大, 张卫军, 牛大鹏. 基于改进 PSO 的发酵过程同步串联混合建模. 自动化学报, 2015, 41(3): 620–630

**DOI** 10.16383/j.aas.2015.c131195

### Simultaneous Series Hybrid Modeling for Fermentation Process Based on Improved Particle Swarm Optimization

YANG Qiang-Da<sup>1</sup> ZHANG Wei-Jun<sup>1</sup> NIU Da-Peng<sup>2</sup>

**Abstract** An accurate and reliable model is the basis and premise for achieving fermentation process optimization. Series hybrid modeling is a relatively more effective method for fermentation process with complex reaction mechanisms, but it needs data obtained from interpolation to develop the black-box models of intermediate variables, which considerably influences the generalization performance of the final hybrid model. Therefore, in this paper, we present a simultaneous series hybrid modeling method, which can transform the black-box model development problem into a dynamic model parameter identification problem, and thus overcome the shortage that existing methods need interpolation data for the development of black-box models. By introducing multi-elite learning and adaptive inertia weight adjustment strategies, an improved particle swarm optimization (PSO) called adaptive multi-elite learning PSO (AMLPSO) is constructed to determine the parameters of black-box models. Uniform design method is used to select the structure of black-box models. Experimental study is carried out based on the practical production data from nosiheptide batch fermentation process, and the results show the effectiveness of the proposed method.

**Key words** Fermentation process, simultaneous series hybrid modeling, particle swarm optimization (PSO), algorithm improvement, uniform design

**Citation** Yang Qiang-Da, Zhang Wei-Jun, Niu Da-Peng. Simultaneous series hybrid modeling for fermentation process based on improved particle swarm optimization. *Acta Automatica Sinica*, 2015, 41(3): 620–630

发酵过程涉及医药、食品、化工、能源和环保等诸多领域, 其在国民经济中的地位日趋重要. 但目前

生产中大都存在产品产量较低、基质转化率和生产效率不高等问题. 通过优化使发酵过程处于最优的环境或操作条件下进行, 是解决上述问题的有效途径<sup>[1]</sup>, 而过程模型是过程优化的基础和前提. 因此, 如何建立准确可靠的过程模型是发酵过程优化的核心问题之一.

现有的发酵过程建模方法大体可分为机理建模方法<sup>[2–3]</sup>、黑箱建模方法<sup>[4–5]</sup>和混合建模方法<sup>[6–7]</sup>三类. 混合建模方法将机理建模与黑箱建模相结合, 通过充分利用已获得的部分先验知识和过程数据, 往往可获得更好的建模效果<sup>[7–9]</sup>. 混合建模方法主要分为并联混合建模方法和串联混合建模方法两种, 前者是利用黑箱模型对机理模型进行附加校正<sup>[7]</sup>, 后者是利用黑箱模型对机理模型中的中间变

收稿日期 2013-12-30 录用日期 2014-11-04  
Manuscript received December 30, 2013; accepted November 4, 2014

高等学校博士学科点专项科研基金 (20120042120014), 国家自然科学基金 (61304121, 51476024), 中央高校基本科研业务费专项资金 (N130404024) 资助

Supported by Specialized Research Fund for the Doctoral Program of Higher Education (20120042120014), National Natural Science Foundation of China (61304121, 51476024), Fundamental Research Funds for the Central Universities (N130404024)

本文责任编辑 胡小平  
Recommended by Associate Editor HU Xiao-Ping

1. 东北大学材料与冶金学院 沈阳 110819 2. 东北大学信息科学与工程学院 沈阳 110819

1. School of Materials and Metallurgy, Northeastern University, Shenyang 110819 2. School of Information Science and Engineering, Northeastern University, Shenyang 110819

量(如菌体比生长速率、基质比消耗速率等)进行估计<sup>[8-9]</sup>. 串联混合建模方法只需要确定机理模型的基本结构, 其涉及的先验知识较并联混合建模方法少, 因而更适于反应机理极其复杂的发酵过程, 且已受到国内外较为广泛的重视和研究<sup>[8-13]</sup>. 然而对于实际发酵过程, 目前尚没有合适的传感器或检测方法能够直接测量中间变量, 所以缺乏构建其黑箱模型的数据对. 针对上述问题, 现有串联混合建模方法大都首先通过插值拟合得到中间变量估算值, 然后利用该估算值完成中间变量黑箱模型的构建, 进而得到发酵过程串联混合模型. 但中间变量估算值的可靠性和准确性往往难以保证, 这无疑会在很大程度上影响所建串联混合模型的泛化性能.

为此, 本文提出一种可将中间变量黑箱模型构建问题转化为动态模型参数辨识问题的同步串联混合建模方法, 以克服现有方法需要利用插值数据来构建黑箱模型的不足; 将多精英学习策略和惯性权重自适应调整策略引入标准粒子群优化 (Particle swarm optimization, PSO) 算法中, 利用前者维持种群的多样性以避免算法陷入局部最优, 利用后者提高算法的搜索效率和收敛速度, 构造了一种改进的 PSO 算法——自适应多精英学习 PSO (Adaptive multi-elite learning PSO, AMLPSO) 算法, 并用之求取黑箱模型的网络参数; 借鉴均匀设计思想确定黑箱模型的网络结构. 最后, 利用诺西肽分批发酵过程实际生产数据对所提方法的性能进行实验研究.

## 1 发酵过程同步串联混合建模

### 1.1 串联混合建模方法的基本思想

依据发酵动力学和质能平衡原理<sup>[13-15]</sup>, 发酵过程机理模型一般可描述为

$$\dot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{x}(t), \boldsymbol{\mu}(t), \mathbf{u}(t)) \quad (1)$$

其中,

$$\boldsymbol{\mu}(t) = \boldsymbol{\gamma}(\mathbf{x}(t), \mathbf{u}(t)) \quad (2)$$

式(1)和式(2)中,  $\mathbf{x} = [x_1, x_2, \dots, x_m]^T$ 、 $\boldsymbol{\mu} = [\mu_1, \mu_2, \dots, \mu_p]^T$  和  $\mathbf{u} = [u_1, u_2, \dots, u_n]^T$  分别表示状态变量、中间变量和操作变量;  $\mathbf{F} = [F_1, F_2, \dots, F_m]^T$  和  $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \dots, \gamma_p]^T$  分别表示相应的函数关系, 其中通过机理建模方法能够获得其具体表达式的, 往往只有前者.

发酵过程串联混合建模方法的基本思想为: 首先确定机理模型的基本结构, 即基于先验知识确定出式(1)的具体表达式; 然后利用黑箱建模方法拟合出函数关系  $\boldsymbol{\gamma}$ , 从而完成串联混合模型的构建.

### 1.2 现有串联混合建模方法

机理模型基本结构一般较易确定, 因此建立串

联混合模型, 关键是中间变量黑箱模型的构建. 然而如上所述, 对于实际发酵过程, 缺乏构建黑箱模型的数据对  $[\mathbf{x}(t), \mathbf{u}(t) | \boldsymbol{\mu}(t)]$ . 为此, 现有方法在构建中间变量黑箱模型时大都采取分步策略: 首先对中间变量进行估算, 即基于状态变量化验值和插值函数获得其导数值, 进而利用式(1)得到中间变量估算值, 并将该估算值作为中间变量黑箱模型的目标输出; 然后建立状态变量、操作变量与中间变量之间的黑箱模型. 该处理方法虽然简单, 但实际发酵过程中状态变量取样周期(通常为几个小时)一般较长, 从而难免对中间变量估算值产生不良影响, 致使其可靠性和准确性难以保证, 这无疑将对串联混合模型的泛化性能产生较大影响. 为此, 提出一种改进的串联混合建模方法, 该方法在构建中间变量黑箱模型时无需对中间变量进行估算. 在某种意义上, 也可将这种改进的串联混合建模方法看作是一种估算建模同步的中间变量黑箱模型构建方法, 因此本文将其称为同步串联混合建模方法.

### 1.3 同步串联混合建模方法

同步串联混合建模方法与现有方法相比, 其不同之处在于它无需利用估算值就可完成中间变量黑箱模型的构建. 下面给出该方法的基本思路.

BP 神经网络除具有较强的非线性映射能力、并行处理能力和自学习能力等神经网络共有特性外, 还具有结构简单、易于实现的特点, 在发酵过程建模领域中得到了较为广泛的应用<sup>[4, 6]</sup>, 故本文选用 BP 神经网络来构建中间变量黑箱模型, 如下式所示.

$$\boldsymbol{\mu}(t) = \boldsymbol{\gamma}_E(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\theta}) \quad (3)$$

其中,  $\boldsymbol{\gamma}_E$  表示用于拟合函数关系  $\boldsymbol{\gamma}$  的基于 BP 神经网络的中间变量黑箱模型;  $\boldsymbol{\theta}$  表示黑箱模型的参数, 即 BP 神经网络的连接权和阈值.

中间变量黑箱模型的构建过程, 就是上述模型参数  $\boldsymbol{\theta}$  的求取过程. 同步串联混合建模方法, 首先把式(3)带入式(1)中, 得到式(4)所示的发酵过程动态模型.

$$\dot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{x}(t), \boldsymbol{\gamma}_E(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\theta}), \mathbf{u}(t)) \quad (4)$$

接着把  $\boldsymbol{\theta}$  看作是动态模型中待辨识的参数, 从而将黑箱模型构建问题转化为动态模型参数辨识问题; 然后以式(5)为目标函数, 基于实际生产数据, 利用合适的优化算法确定出使目标函数值最小的  $\boldsymbol{\theta}$ , 从而完成中间变量黑箱模型的构建, 并避免了需要利用中间变量估算值来构建其黑箱模型的不足.

$$J(\boldsymbol{\theta}) = \frac{\sum_{b=1}^B \sum_{h=1}^{H_b} \sum_{m=1}^M \left| \frac{x_{mbh} - \hat{x}_{mbh}}{x_{mbh}} \right|}{M \cdot \sum_{b=1}^B H_b} \quad (5)$$

其中,  $b = 1, 2, \dots, B$  表示第  $b$  批训练数据, 而  $B$  为训练数据所包含的批次总数;  $h = 1, 2, \dots, H_b$  表示状态变量的第  $h$  个化验值, 而  $H_b$  为第  $b$  批训练数据所包含的状态变量化验值总数;  $m = 1, 2, \dots, M$  表示第  $m$  个状态变量, 而  $M$  为状态变量总数;  $x_{mbh}$  和  $\hat{x}_{mbh}$  分别表示状态变量的化验值和模型预报值.

## 2 PSO 算法及其改进

### 2.1 标准 PSO

PSO 算法由 Kennedy 和 Eberhart 首次提出<sup>[16]</sup>, 具有概念简单、易于实现和收敛速度快等优点, 目前已被广泛应用于多个领域<sup>[17-21]</sup>.

假设在  $D$  维搜索空间中, 粒子  $i$  在第  $g$  代的位置和速度分别为  $\mathbf{z}_i^g = (z_{i1}^g, z_{i2}^g, \dots, z_{iD}^g)$  和  $\mathbf{v}_i^g = (v_{i1}^g, v_{i2}^g, \dots, v_{iD}^g)$ , 其所经历的最好位置为  $\mathbf{pbest}_i = (pbest_{i1}, pbest_{i2}, \dots, pbest_{iD})$ , 整个种群所经历的最好位置为  $\mathbf{gbest} = (gbest_1, gbest_2, \dots, gbest_D)$ . 则在标准 PSO 算法中, 对于第  $g+1$  代, 各个粒子分别依据式 (6) 和式 (7) 来更新其速度和位置的第  $d$  维.

$$v_{id}^{g+1} = \omega \cdot v_{id}^g + c_1 \cdot r_1 (pbest_{id} - z_{id}^g) + c_2 \cdot r_2 (gbest_d - z_{id}^g) \quad (6)$$

$$z_{id}^{g+1} = z_{id}^g + v_{id}^{g+1} \quad (7)$$

式 (6) 中,  $\omega$  为惯性权重;  $c_1$  和  $c_2$  为加速常数;  $r_1$  和  $r_2$  为服从均匀分布  $U(0, 1)$  的随机数.

### 2.2 自适应多精英学习 PSO

标准 PSO 算法容易陷入局部最优, 尤其是当目标函数维数较高或优化问题较复杂时, 它陷入局部最优的概率更大<sup>[22]</sup>. 为此, 提出一种多精英学习策略, 以维持种群的多样性, 使算法避免陷入局部最优. 同时还提出一种惯性权重自适应调整策略, 以提高算法的搜索效率、加快其收敛速度. 最后将上述两种策略引入标准 PSO 算法中, 构造了一种自适应多精英学习 PSO (AMLPSO) 算法.

#### 2.2.1 多精英学习策略

多精英学习策略是基于如下认识而提出的: 1) 维持种群多样性是避免 PSO 算法陷入局部最优的有效手段<sup>[23]</sup>, 而采用多精英学习策略代替标准 PSO 算法中的单精英 (即  $\mathbf{gbest}$ ) 学习策略, 无疑将有助于维持种群的多样性; 2) 在社会实践活动中, 一个人同时向多个精英学习所获得的效果往往比其仅仅向一个精英学习要好得多. 据此, 多精英学习策略将式 (6) 所示的速度更新公式改进为如下形式:

$$v_{id}^{g+1} = \omega \cdot v_{id}^g + c_1 \cdot r_1 (pbest_{id} - z_{id}^g) + c_2 \cdot r_2 (elite_{r_3d} - z_{id}^g) \quad (8)$$

其中,  $elite_{r_3d}$  为  $elite_{r_3}$  的第  $d$  维, 而  $elite_{r_3}$  选自精英集合  $E$ ;  $r_3$  为  $1 \sim S_E$  之间的随机整数, 而  $S_E$  表示精英集合  $E$  所包含的精英个数. 精英集合  $E$  的搭建和更新过程如算法 1 所述.

#### 算法 1. 精英集合 $E$ 的搭建与更新

**情况 1.** 若精英集合  $E$  是空的, 则按适应度函数值的优劣对  $\mathbf{pbest}_i$  ( $i = 1, 2, \dots, S$ ) 进行排序, 把前  $S_E$  个较优的  $\mathbf{pbest}$  加入  $E$  中, 同时将其记为  $elite_j$  ( $j = 1, 2, \dots, S_E$ ), 并退出.

**情况 2.** 若精英集合  $E$  是非空的, 且存在劣于  $\mathbf{gbest}$  的  $elite \in E$ , 则用  $\mathbf{gbest}$  替代  $E$  中最劣的那个精英, 并退出.

**情况 3.** 若精英集合  $E$  是非空的, 且不存在劣于  $\mathbf{gbest}$  的  $elite \in E$ , 则保持  $E$  不变, 并退出.

由式 (8) 可以看出, 多精英学习策略能够使各个粒子同时向多个精英学习, 且同一精英对不同粒子具有不同的影响, 这样就维持了种群在迭代过程中的多样性, 从而避免 AMLPSO 算法陷入局部最优, 使该算法具有较强的全局搜索能力.

#### 2.2.2 惯性权重自适应调整策略

为了提高算法的搜索效率和收敛速度, 提出一种惯性权重自适应调整策略, 根据每个粒子前后两代适应度函数值的变化情况, 独立地对其惯性权重进行自适应调整, 如式 (9) 和式 (10) 所示. 该调整策略的基本出发点为, 如果粒子在当前速度驱使下能够获得一个较其先前位置更优的位置, 则应加大该粒子“惯性”部分的影响, 即增加其惯性权重; 反之则减小其惯性权重.

$$\omega_i^g = \omega_{\min} + \frac{1}{1 + e^{-\frac{\Delta J_i^g}{\alpha}}} (\omega_{\max} - \omega_{\min}) \quad (9)$$

其中,

$$\Delta J_i^g = \begin{cases} J_i^{g-1} - J_i^g, & \text{所求问题为极小化问题} \\ J_i^g - J_i^{g-1}, & \text{所求问题为极大化问题} \end{cases} \quad (10)$$

式 (9) 和式 (10) 中,  $\omega_i^g$  为粒子  $i$  在第  $g$  代的惯性权重;  $\omega_{\min}$  和  $\omega_{\max}$  分别为惯性权重的下限值和上限值;  $\alpha$  为正的常数;  $J_i^g$  和  $J_i^{g-1}$  分别为粒子  $i$  在第  $g$  代和第  $g-1$  代的适应度函数值.

由式 (9) 和式 (10) 可以看出, 惯性权重自适应调整策略能够增大潜在优良惯性方向的影响, 同时减小潜在平庸惯性方向的影响, 这样就在一定程度上减少了算法搜索的盲目性, 使其搜索效率和收敛速度得以提高.

需要说明的是, 当  $g=0$  时, 各个  $\omega_i$  均为  $[\omega_{\min}, \omega_{\max}]$  区间内的随机数; 当  $g \geq 1$  时, 各个  $\omega_i$  才依据式 (9) 和式 (10) 经计算而得出.

### 2.2.3 AMLPSO 算法步骤

#### 算法 2. AMLPSO 算法

##### 步骤 1. 初始化:

1) 设定待求问题维数  $D$ 、种群规模  $S$ 、精英集合大小  $S_E$ 、参数  $\alpha$ 、惯性权重最小值  $\omega_{\min}$  和最大值  $\omega_{\max}$ 、加速常数  $c_1$  和  $c_2$ 、最大迭代次数  $Gen$ 。

2) 在待求问题解空间中, 随机生成  $S$  个初始位置和速度分别为  $\mathbf{z}_i^0 = (z_{i1}^0, z_{i2}^0, \dots, z_{iD}^0)$  和  $\mathbf{v}_i^0 = (v_{i1}^0, v_{i2}^0, \dots, v_{iD}^0)$  的粒子, 并由此组成初始种群。

3) 计算各个粒子的适应度函数值, 并将其记为  $J_i^g (i = 1, 2, \dots, S)$ , 其中  $g = 0$ 。

4) 令  $\mathbf{pbest}_i = \mathbf{z}_i^0$ , 并取  $\mathbf{pbest}_i (i = 1, 2, \dots, S)$  中适应度函数值最优者为  $\mathbf{gbest}$ 。

5) 利用算法 1 搭建精英集合  $E$ 。

6) 令迭代次数  $g = 0$ 。

**步骤 2.** 若  $g \geq Gen$ , 则输出  $\mathbf{gbest}$ , 并退出; 否则, 执行以下步骤:

1) 对于每个粒子 ( $i = 1, 2, \dots, S$ ), 循环执行步骤 a)~f)。

a) 计算粒子  $i$  的惯性权重. 若  $g = 0$ , 则在  $[\omega_{\min}, \omega_{\max}]$  区间内产生一个随机数, 并将该随机数赋值给惯性权重  $\omega_i^g (g = 0)$ ; 若  $g \geq 1$ , 则依据式 (9) 和式 (10) 经计算得出惯性权重  $\omega_i^g$ 。

b) 更新粒子  $i$  的速度和位置. 首先令式 (8) 中的  $\omega$  等于  $\omega_i^g$ , 然后依据式 (8) 和式 (7) 更新粒子  $i$  的速度和位置, 并分别记为  $\mathbf{v}_i^{g+1}$  和  $\mathbf{z}_i^{g+1}$ 。

c) 计算粒子  $i$  的适应度函数值, 并将其记为  $J_i^{g+1}$ 。

d) 更新  $\mathbf{pbest}_i$ . 若  $\mathbf{z}_i^{g+1}$  较  $\mathbf{pbest}_i$  所对应的适应度函数值更优, 则  $\mathbf{pbest}_i = \mathbf{z}_i^{g+1}$ ; 否则, 保持  $\mathbf{pbest}_i$  不变。

e) 更新  $\mathbf{gbest}$ . 若  $\mathbf{pbest}_i$  较  $\mathbf{gbest}$  所对应的适应度函数值更优, 则  $\mathbf{gbest} = \mathbf{pbest}_i$ ; 否则, 保持  $\mathbf{gbest}$  不变。

f) 利用算法 1 更新精英集合  $E$ 。

2) 令  $g = g + 1$ 。

## 3 基于 AMLPSO 的发酵过程同步串联混合建模

如第 1.2 节所述, 建立发酵过程串联混合模型, 关键是中间变量黑箱模型的构建. 而要利用同步串联混合建模方法完成中间变量黑箱模型的构建, 主要需解决如下两个问题: 1) 求解式 (4) 所示动态模型的参数辨识问题, 以获得黑箱模型网络参数  $\theta$ ; 2) 确定黑箱模型网络结构. 下面给出针对上述两个问题的解决方案。

### 3.1 黑箱模型网络参数 $\theta$ 的确定

式 (4) 所示动态模型的参数辨识问题是一个高

维复杂优化问题, 而 AMLPSO 算法是为克服标准 PSO 算法在求解优化问题 (尤其是高维复杂优化问题) 时易陷入局部最优这一不足而提出的, 且具有较高的搜索效率和收敛速度. 因此, 选用 AMLPSO 算法求解上述动态模型参数辨识问题, 从而获得黑箱模型的网络参数  $\theta$ . 在求解过程中, 每个粒子的位置代表参数  $\theta$  的一组候选解, 并以式 (5) 为适应度函数来评价粒子的优劣. 对于粒子  $i$ , 其第  $g$  代适应度函数值的计算过程如算法 3 所述。

#### 算法 3. 粒子 $i$ 适应度函数值的计算

**步骤 1.** 令适应度函数值  $J_i^g = 0$ 。

**步骤 2.** 令批次次数  $b = 1$ 。

**步骤 3.** 若  $b > B$  (其中  $B$  表示训练数据所包含的批次总数), 则转步骤 4; 否则, 执行以下步骤:

1) 将粒子  $i$  所对应的参数  $\theta$  的值带入式 (4) 所示动态模型, 然后基于第  $b$  批状态变量初始值和操作变量测量值, 采用四阶 Runge-Kutta 方法以操作变量采样周期为步长对该动态模型进行求解, 进而得到状态变量的预报值  $\hat{\mathbf{x}}_{bh} = [\hat{x}_{1bh}, \hat{x}_{2bh}, \dots, \hat{x}_{Mbh}] (h = 1, 2, \dots, H_b)$ 。

2) 计算该动态模型对第  $b$  批状态变量预报的相对误差和:  $Error_b = \sum_{h=1}^{H_b} \sum_{m=1}^M | \frac{x_{mbh} - \hat{x}_{mbh}}{x_{mbh}} |$ 。

3) 令  $J_i^g = J_i^g + Error_b$ 。

4) 令  $b = b + 1$ 。

**步骤 4.** 令  $J_i^g = \frac{J_i^g}{M \cdot \sum_{b=1}^B H_b}$ 。

采用 AMLPSO 算法确定黑箱模型网络参数  $\theta$  的具体流程如图 1 所示。

### 3.2 黑箱模型网络结构的确定

黑箱模型网络结构是指构建中间变量黑箱模型的 BP 神经网络的结构 (包括输入层节点数、输出层节点数、隐含层层数以及各隐含层节点数), 其合理选取对于获得高性能的过程模型至关重要. 单隐含层 BP 神经网络可以简化网络结构的确定过程, 且能以任意精度逼近任意非线性连续函数<sup>[24]</sup>, 故用之构建中间变量黑箱模型. 一般地, 输入层和输出层的节点数是由实际问题而定的, 这样最终可将黑箱模型网络结构的确定归结为模型隐含层节点数的确定。

考虑到如下两点: 1) 各个中间变量的影响因素可能有所不同, 因而在构建中间变量黑箱模型时, 各个中间变量所需的模型输入可能不同; 2) 即便影响因素相同, 各个中间变量黑箱模型的最佳网络结构也可能不同, 本文为各个中间变量分别构建独立的黑箱模型. 但这会使确定网络结构的工作量大幅增加, 其原因是这样的: BP 神经网络隐含层节点数的选取, 目前尚无理论上的指导, 大多根据经验采用凑试法确定. 假设中间变量的个数为  $p$ , 其黑箱模型隐含层节点数分别有  $Hid_1, Hid_2, \dots, Hid_p$  种选择, 这样要确定出一组最佳的网络结构, 就需进行

$Hid_1 \times Hid_2 \times \dots \times Hid_p$  次凑试 (之所以进行如此多的凑试, 是因为混合模型性能受到各个黑箱模型的综合影响, 很难单独分析各个黑箱模型对混合模型的影响). 很显然, 当中间变量个数相对较多, 且其黑箱模型隐含层节点数的可选范围相对较宽时, 确定黑箱模型网络结构将面临巨大的工作量.

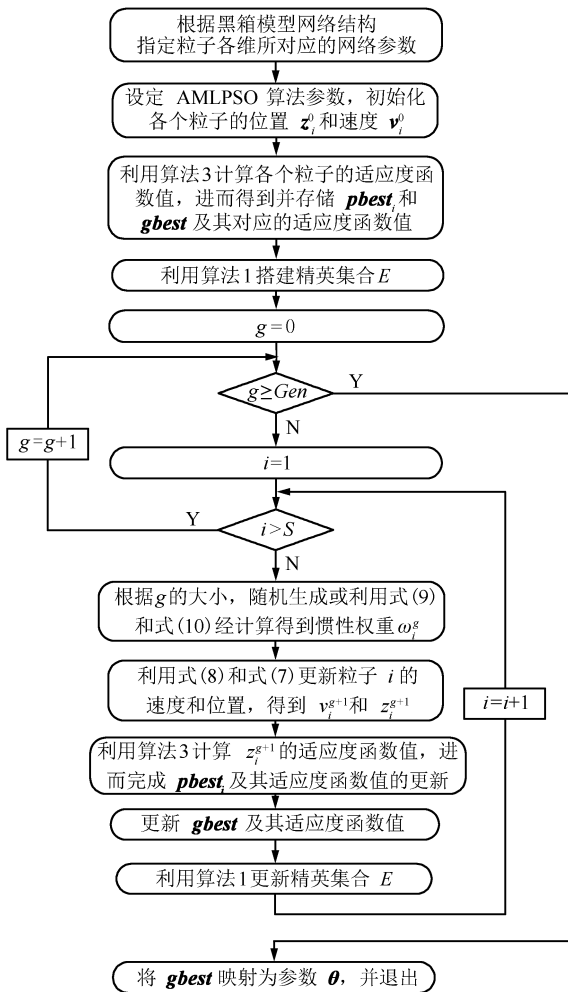


图1 采用 AMLPSO 算法确定参数  $\theta$  的流程图

Fig. 1 The flow chart of determination of parameter ( $\theta$ ) by using AMLPSO algorithm

均匀设计是科学设计多因素多水平试验的一种方法, 能在不影响试验效果的前提下, 有效减少试验的次数<sup>[25]</sup>, 因此可用于解决上述黑箱模型网络结构的确定问题. 其基本思想为: 首先以各个黑箱模型为元素, 黑箱模型隐含层节点数为水平建立均匀设计表; 然后以留一 (这里的“一”是指一批) 交叉验证所得的平均相对验证误差为指标, 完成黑箱模型网络结构的确定.

综上所述, 基于 AMLPSO 的发酵过程同步串联混合建模的基本步骤为: 首先利用机理建模方法确定出机理模型的基本结构; 接着将中间变量黑箱模型构建问题转化为动态模型参数辨识问题; 然后

采用 AMLPSO 算法求解上述动态模型参数辨识问题, 从而获得黑箱模型的网络参数, 并借鉴均匀设计思想确定出黑箱模型的网络结构.

## 4 基于诺西肽发酵过程的实验研究

诺西肽 (Nosiheptide) 是一种动物饲料添加剂<sup>[26]</sup>, 具有广阔的市场应用前景. 但目前工业生产中存在产物浓度较低、基质转化率和生产效率不高等问题. 对该发酵过程进行优化是解决这些问题的一条有效途径, 而过程模型是过程优化的基础和前提. 因此, 研究高性能过程模型的构建方法具有十分重要的实际意义. 本文以某大学“211 工程”综合自动化实验室中的诺西肽发酵过程为对象, 对前述所提方法进行实验研究.

### 4.1 实验数据

实验中采取分批发酵方式, 每批诺西肽发酵过程的持续时间为 96 h 左右. 从 7 批正常发酵所得的实验数据中, 随机选取 6 批用于构建过程串联混合模型, 剩余 1 批用于测试所建模型的性能. 其中, 状态变量 (包括菌体浓度、基质浓度和产物浓度) 数据是通过人工取样送化验室分析化验获得, 其取样周期为 3 h 左右; 操作变量 (包括温度、pH 值、溶解氧浓度等) 数据是通过 DCS 系统获得, 其采样周期为 5 min.

### 4.2 过程串联混合模型的构建

#### 4.2.1 机理模型基本结构

依据诺西肽分批发酵过程实际情况和相关研究成果<sup>[14-15, 27-28]</sup>, 其机理模型基本结构可描述为

$$\begin{cases} \dot{x}_1 = \mu_1 x_1 = \gamma_1 (x_1, x_2, u_1, u_2, u_3) x_1 \\ \dot{x}_2 = -\mu_2 x_2 = -\gamma_2 (x_1, x_2, x_3, u_1, u_2, u_3) x_2 \\ \dot{x}_3 = \mu_3 x_3 - \beta x_3 = \gamma_3 (x_2, u_1, u_2, u_3) x_3 - \beta x_3 \end{cases} \quad (11)$$

其中,  $x_1$ 、 $x_2$ 、 $x_3$  为过程状态变量 (g/L), 分别表示菌体浓度、基质浓度和产物浓度;  $\mu_1$ 、 $\mu_2$ 、 $\mu_3$  为过程中间变量 ( $h^{-1}$ ), 分别表示菌体比生长速率、基质比消耗速率和产物比生成速率;  $u_1$ 、 $u_2$ 、 $u_3$  为过程操作变量, 分别表示温度 ( $^{\circ}C$ )、pH 值和溶解氧浓度 (g/L);  $\beta$  表示产物水解速率常数, 经实验知其大小为  $0.0004 h^{-1}$ ;  $\gamma_1$ 、 $\gamma_2$ 、 $\gamma_3$  表示相应的函数关系, 但其具体表达式未知.

#### 4.2.2 中间变量黑箱模型

机理模型基本结构确定后, 利用第 3 节所述方法为中间变量  $\mu_1$ 、 $\mu_2$  和  $\mu_3$  建立如图 2 所示的黑箱模型 (图中,  $input = -1$  是为引入阈值而设置的), 进而完成诺西肽分批发酵过程串联混合模型的构建, 其中相关参数的设置情况如下所述.

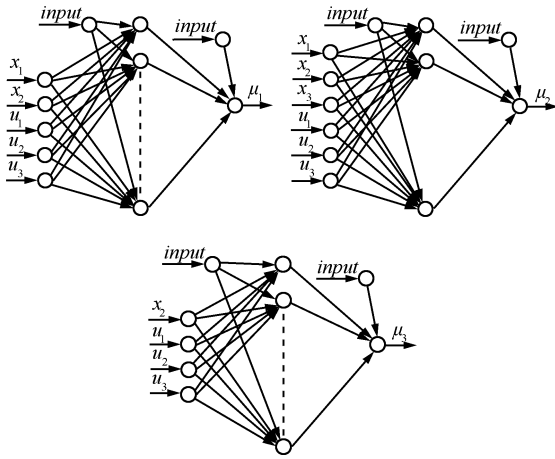


图 2 基于 BP 神经网络的中间变量黑箱模型

Fig. 2 The black box-models of intermediate variables based on BP neural network

在 AMLPSO 算法中, 每个粒子的位置代表 3 个黑箱模型连接权和阈值的一组候选解, 因此其维数  $D = [(5 + 1) \times hid_1 + (hid_1 + 1) \times 1] + [(6 + 1) \times hid_2 + (hid_2 + 1) \times 1] + [(4 + 1) \times hid_3 + (hid_3 + 1) \times 1]$ , 其中  $hid_1$ 、 $hid_2$ 、 $hid_3$  分别表示 3 个黑箱模型隐含层的节点数; 惯性权重的最小值  $\omega_{min} = 0.4$  和最大值  $\omega_{max} = 0.9$ ; 加速常数  $c_1 = c_2 = 2$ ; 种群规模  $S = 100$ ; 精英集合大小  $S_E$  和参数  $\alpha$  通过试验的方法确定。

1) 精英集合大小  $S_E$  的确定

从表 1 中随机选取 3 组黑箱模型网络结构用于研究精英集合大小  $S_E$  对算法性能的影响. 它们分别是第 3 组、第 5 组和第 17 组. 针对各组网络结构,  $S_E$  取值分别为 5、10、...、50 的多精英学习 PSO 算法 (仅将本文所提出的多精英学习策略引入标准 PSO 算法中) 均基于 6 批构建过程模型的实验数据独立重复运行 20 次 (最大迭代次数  $Gen = 2000$ ), 其平均最优目标函数值如图 3 所示. 本文引入多精英学习策略的主要目的是提高算法的全局搜索能力, 且所涉及的优化问题是一个极小化问题, 因而平均最优目标函数值越小越好. 而由图 3 可以看出, 当  $S_E = 20$  时可取得较好的优化效果, 因此本文将精英集合大小  $S_E$  设定为 20.

2) 参数  $\alpha$  的确定

以算法达到指定寻优精度所需要的迭代次数为指标, 利用上述 3 组黑箱模型网络结构来确定参数  $\alpha$  的取值. 本文为第 3 组、第 5 组和第 17 组网络结构指定的寻优精度分别为 0.15、0.03 和 0.1. 针对各组网络结构,  $\alpha$  取值分别为 0.01、0.02、...、0.1 的 AMLPSO 算法均基于 6 批构建过程模型的实验数据独立重复运行 20 次 (算法终止条件为达到指定寻优精度或迭代次数达到 2000 次, 精英集合大小  $S_E = 20$ ), 其达到指定寻优精度所需迭代次数的中

位值如图 4 所示 (当迭代次数达到 2000 次但仍未获得指定寻优精度时, 认为本次寻优过程失败, 将与之对应的所需迭代次数记为无穷大, 因此用中位值代替平均值来完成参数  $\alpha$  的确定). 本文引入惯性权重自适应调整策略的主要目的是提高算法的收敛速度, 因而达到指定寻优精度所需的迭代次数越少越好. 而由图 4 可以看出, 当  $\alpha$  取值为 0.04、0.05 或 0.06 时均可获得较好的收敛性能, 本文将参数  $\alpha$  设定为 0.05.

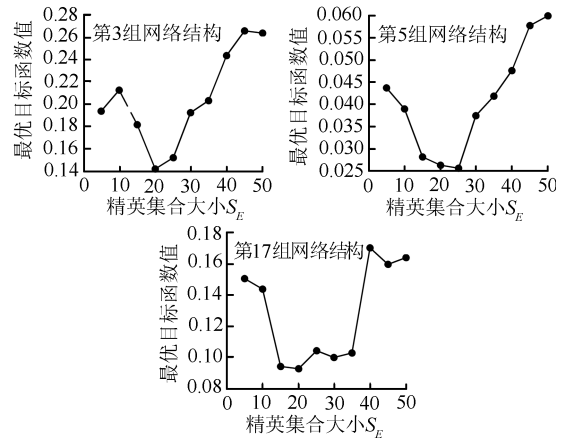
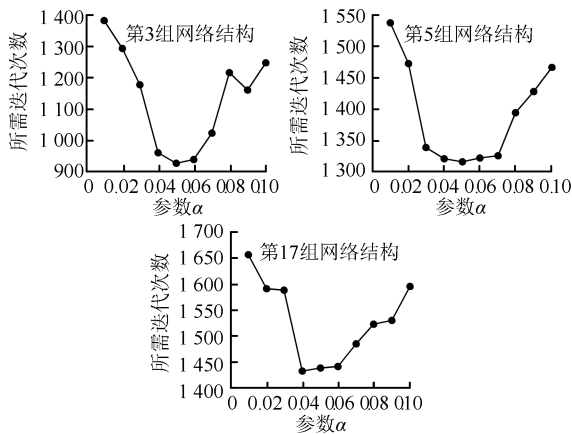


图 3 基于不同  $S_E$  的试验结果

Fig. 3 The test results with different  $S_E$

在 3 个中间变量黑箱模型中, 隐含层节点的转移函数均选用 Sigmoid 函数, 输出层节点的转移函数均选用 Purelin 函数. 应用均匀设计思想确定各个黑箱模型隐含层的节点数, 其具体步骤为: 首先依据以往经验将各个黑箱模型隐含层的节点数选定在 [6, 25] 范围内; 然后以各个黑箱模型为元素, 黑箱模型隐含层的节点数为水平, 根据文献 [25] 构造如表 1 所示的均匀设计表  $U_{20}(20^3)$ , 并按照该表的水平组合进行试验, 记录各组水平组合下留一交叉验证所得的平均相对验证误差 (Mean relative error of validation, MREV) (其中 MREV 的计算过程为, 把 6 批构建过程模型的实验数据分成两部分: 1) 是除去某批实验数据后所有实验数据的集合 (共含 5 批实验数据), 将这部分实验数据作为训练集并基于所选水平组合和第 3 节所述建模方法构建一个串联混合模型; 2) 是刚才被排除的那批实验数据, 将这批实验数据作为验证集对前面所建串联混合模型的性能进行验证, 并计算得到模型对状态变量的相对验证误差. 该过程重复 6 次, 直到所有批次的实验数据均作过一次验证集, 这样总共可得到 6 批相对验证误差, 取其均值作为 MREV). 为了获得有效的统计信息, 每组水平组合均独立重复试验 20 次, 并取这 20 次试验所得的平均相对验证误差的平均值 (Mean MREV, MMREV) 为最终试验结果, 同时将其列于表 1 的最后 1 列; 最后从各组试验中挑

图4 基于不同  $\alpha$  的试验结果Fig. 4 The test results with different  $\alpha$ 表1  $U_{20}(20^3)$  均匀设计表及试验结果Table 1 The uniform design table of  $U_{20}(20^3)$  and experimental results

组	菌体比生长速率黑箱模型	基质比消耗速率黑箱模型	产物比生成速率黑箱模型	MMREV (%)
1	6	14	20	19.71
2	7	22	13	15.33
3	8	8	9	20.15
4	9	20	24	13.36
5	10	12	16	4.73
6	11	17	6	17.94
7	12	7	22	14.78
8	13	25	17	11.27
9	14	10	11	3.69
10	15	18	19	8.16
11	16	24	8	12.49
12	17	11	25	11.83
13	18	15	12	4.08
14	19	19	15	6.65
15	20	6	14	18.56
16	21	23	21	9.44
17	22	13	7	13.69
18	23	16	23	10.34
19	24	21	10	11.02
20	25	9	18	14.43

选一组 MMREV 最小的水平组合, 并将其作为所要求取的最佳组合. 由表 1 可以看出, 第 5、9、13 组水平组合均取得了较好建模效果, 尤其是第 9 组水平组合的建模效果更为理想, 因此将该组水平组合作为黑箱模型隐含层节点数的最佳组合, 即将菌体比生长速率、基质比消耗速率和产物比生成速率黑箱模型的网络结构分别确定为  $5 \times 14 \times 1$ 、 $6 \times 10 \times 1$  和  $4 \times 11 \times 1$ . 另外, 由表 1 还可以看出, 第 5、9、13 组水平组合之间有较明显的差异, 说明各个黑箱模

型之间是相互耦合的, 所以利用同步串联混合建模方法构建发酵过程模型时, 应当综合考虑各个黑箱模型隐含层节点数的选取问题.

中间变量黑箱模型的网络结构确定之后, 将 6 批构建过程模型的实验数据均作为训练集, 并利用第 3 节所述建模方法获得中间变量黑箱模型的网络参数, 进而完成诺西肽分批发酵过程串联混合模型的构建.

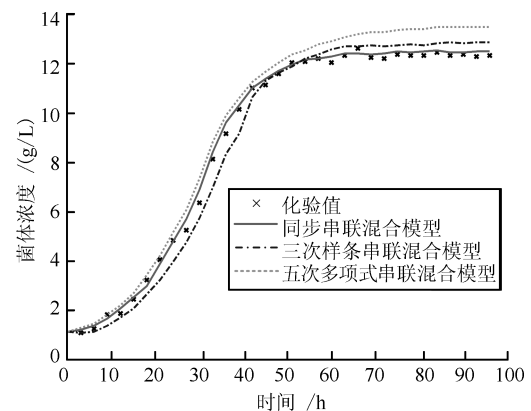
#### 4.2.3 结果及分析

将测试批次实验数据中的状态变量初始值和操作变量测量值输入上述串联混合模型中, 可得到该模型对诺西肽分批发酵过程中菌体浓度、基质浓度和产物浓度 3 个状态变量的预报情况, 如图 5 所示.

由图 5 中菌体浓度、基质浓度和产物浓度的化验值与串联混合模型的输出值曲线可知, 基于 AMLPSO 的同步串联混合建模方法所构建的诺西肽分批发酵过程模型能以较高精度对该过程中的上述 3 个状态变量进行预报. 预报结果的平均相对误差 (Mean relative error, MRE) 分别为 3.16%、3.28% 和 4.06%, 均方根误差 (Root mean square error, RMSE) 分别为 0.2105 g/L、0.6106 g/L 和 0.0147 g/L, 可以满足过程优化的需要. 由上述实验结果可以看出, 基于 AMLPSO 的发酵过程同步串联混合建模方法是有效的.

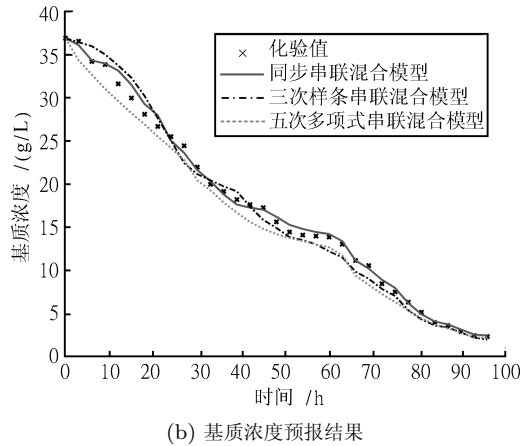
#### 4.3 对比实验

为体现基于 AMLPSO 的同步串联混合建模方法的优势, 做如下对比实验: 1) 对比同步串联混合建模方法与现有方法的建模效果; 2) 对比标准 PSO (Standard PSO, SPSO)、自适应 PSO (Adaptive PSO, APSO) (仅将惯性权重自适应调整策略引入 SPSO 中)、多精英学习 PSO (Multi-elite learning PSO, MLPPO) (仅将多精英学习策略引入 SPSO

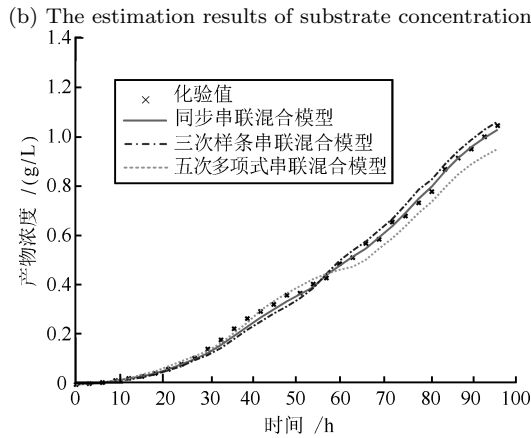


(a) 菌体浓度预报结果

(a) The estimation results of biomass



(b) 基质浓度预报结果



(c) 产物浓度预报结果

(c) The estimation results of production concentration

图 5 状态变量预报结果

Fig. 5 The estimation results of state variables

中) 和 AMLPSO 的寻优效果, 并进一步对比混沌 PSO (Chaotic PSO, CPSO)<sup>[17]</sup>、量子行为 PSO (Quantum-behaved PSO, QPSO)<sup>[29]</sup>、禁忌搜索 (Tabu search, TS)<sup>[30]</sup>、遗传算法 (Genetic algorithm, GA) 和 AMLPSO 的寻优效果。

#### 4.3.1 建模效果的对比

为体现同步串联混合建模方法的优势, 本文还利用现有串联混合建模方法为诺西肽分批发酵过程构建过程模型, 基本步骤为: 首先基于上述 6 批实验数据中的状态变量化验值和插值函数 (本文采用在发酵过程串联混合建模中常用的两种插值函数: 三次样条插值函数<sup>[12]</sup>和五次多项式插值函数<sup>[13]</sup>) 获得状态变量的导数值, 进而利用式 (11) 得到中间变量的估算值; 然后选用单隐含层 BP 神经网络为菌体比生长速率、基质比消耗速率和产物比生成速率分别构建网络结构为  $5 \times 14 \times 1$ 、 $6 \times 10 \times 1$  和  $4 \times 11 \times 1$  的黑箱模型, 并采用 AMLPSO 算法 (算法参数的设置同第 4.2.2 节) 求取上述 3 个中间变量黑箱模型的网络参数 (需要说明的是, 由于中间变量黑箱模型的目标输出已经通过估算的方法获得, 因此这里可

以采用常规的 BP 神经网络训练算法来完成各个黑箱模型的构建. 但通过实验发现, AMLPSO 算法较之常规的 BP 神经网络训练算法, 其所建过程模型的性能更优, 因而本文仅给出采用 AMLPSO 算法所建过程模型的预报结果), 完成基于现有串联混合建模方法的诺西肽分批发酵过程模型的构建。

图 5 中的点划线和虚线分别给出了上述两个基于现有串联混合建模方法的诺西肽分批发酵过程模型对测试批次发酵过程菌体浓度、基质浓度和产物浓度 3 个状态变量的实验结果. 表 2 示出了应用上述两种串联混合建模方法所构建的 3 个诺西肽分批发酵过程模型 (同步串联混合建模方法所构建的过程模型记为 Model<sub>1</sub>; 基于三次样条和五次多项式插值的现有串联混合建模方法所构建的过程模型分别记为 Model<sub>2</sub> 和 Model<sub>3</sub>) 对测试批次实验数据中菌体浓度、基质浓度和产物浓度 3 个状态变量预报的平均相对误差 (Mean relative error, MRE) 和均方根误差 (Root mean square error, RMSE)。

由图 5 可以看出, 上述两种串联混合建模方法所构建的诺西肽分批发酵过程模型均能预报该发酵过程中菌体浓度、基质浓度和产物浓度 3 个状态变量的变化趋势. 由表 2 中的数据可以清楚地看出, 同步串联混合建模方法所构建的诺西肽分批发酵过程模型的泛化性能明显好于现有串联混合建模方法, 表明在发酵过程建模中同步串联混合建模方法较之现有串联混合建模方法具有显著的泛化性能优势. 但同步串联混合建模方法需要花费相对较多的计算时间, 例如基于相同的机器配置和编译平台 (实验所用工作站的配置为 CPU Xeon E5-2620 V2 2.1 GHz 2 颗, 内存 32 GB, 操作系统 Windows 7, 编译平台为 Matlab R2009b), 构建上述混合模型 Model<sub>1</sub>、Model<sub>2</sub> 和 Model<sub>3</sub> 所花费的计算时间分别为 547.3915 s、16.9314 s 和 17.6806 s (20 次测试结果的平均值). 下面从时间复杂度的角度来分析其具体原因. 对于现有和同步串联混合建模方法而言, 其构建过程模型的时间花费均主要用于神经网络的计算 (本文将给定一组输入计算其输出, 称为一次神经网络计算), 因此在对两种建模方法的时间复杂度进行分析时仅考虑神经网络的计算时间. 由上述建模过程可知, 现有串联混合建模方法的时间复杂度大体可表示为  $33$  (每批发酵过程可得 33 个取样点)  $\times 3$  (中间变量个数)  $\times 6$  (批次)  $\times 100$  (种群规模)  $\times 2000$  (迭代次数)  $\times t$  (单次神经网络计算时间); 同步串联混合建模方法的时间复杂度大体可表示为  $1152$  (本文以操作变量采样周期为步长进行离散化, 这样每批发酵过程大约需要进行 1152 次神经网络计算)  $\times 3$  (中间变量个数)  $\times 6$  (批次)  $\times 100$  (种群规模)  $\times 2000$  (迭代次数)  $\times t$  (单次神经网络计算时间). 由此可见, 在以操作变量采样周期为步长进



行离散化的前提下,同步串联混合建模方法的时间复杂度是现有方法的 34.9 倍(与上述测试结果基本相符).另外,需要说明的是,如果拟减少同步串联混合建模方法的时间复杂度,则可以通过增加离散步长的方法来实现(比方说将步长变为操作变量采样周期的 4 倍,那么同步串联混合建模方法的时间复杂度将变为现有方法的 8.7 倍),但这无疑会在一定程度上影响所建串联混合模型的泛化性能.

表 2 状态变量预报误差

Table 2 The estimation errors of state variables

模型	菌体浓度		基质浓度		产物浓度	
	MRE	RMSE	MRE	RMSE	MRE	RMSE
	(%)	(g/L)	(%)	(g/L)	(%)	(g/L)
Model <sub>1</sub>	3.16	0.2105	3.28	0.6106	4.06	0.0147
Model <sub>2</sub>	6.66	0.5096	6.91	1.1305	7.92	0.0279
Model <sub>3</sub>	8.14	0.7455	8.45	1.4243	9.31	0.0391

综上所述,与现有串联混合建模方法相比,同步串联混合建模方法具有更好的泛化性能,但其计算时间会有一定程度的增加.然而对于过程离线建模而言,建模效果较之计算时间相对更为重要,因此对于模型预报精度可以提高 48.7% 以上的建模效果而言,多花费不超过 10 min 的计算时间是值得的.

#### 4.3.2 寻优效果的对比

为了体现 AMLPSO 算法性能及其对 PSO 算法的改进效果,分别利用 SPSO、APSO、MLPSO 和 AMLPSO 并基于上述 6 批构建过程模型的实验数据,求取诺西肽分批发酵过程同步串联混合建模中 3 个中间变量黑箱模型的网络参数.其中相关参数的设置情况为:3 个黑箱模型的网络结构同第 4.2.2 节;SPSO、APSO、MLPSO 和 AMLPSO 四种算法中,SPSO 和 MLPSO 的惯性权重在迭代过程中保持 0.729 不变,其他算法参数的设置同第 4.2.2 节.每种优化算法均独立重复运行 20 次,其平均最优适应度函数值的收敛曲线如图 6 所示.

由图 6 可以看出,APSO 算法相比 SPSO 算法具有更快的收敛速度,且寻优精度也有所提高;MLPSO 算法相比 SPSO 算法具有更高的寻优精度,但收敛速度相对较慢;AMLPSO 算法较 SPSO 算法具有明显的寻优精度优势和大致相当的收敛速度.上述实验结果表明,惯性权重自适应调整策略可以提高算法搜索效率和收敛速度;多精英学习策略可以提高算法全局搜索能力,防止算法陷入局部最优;而 AMLPSO 算法通过引入上述两种策略,可以在提高算法全局搜索能力的同时获得较快的收敛速度,从而得到较好的寻优效果.

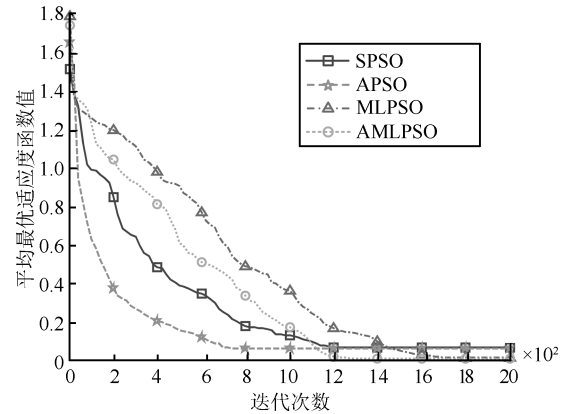


图 6 平均最优适应度函数值收敛曲线

Fig. 6 The convergence curves of mean best fitness function values

为了进一步体现 AMLPSO 算法在求解高维复杂优化问题时的性能,还基于 Sphere ( $f_1$ )、Rastrigrin ( $f_2$ ) 和 Griewank ( $f_3$ ) 3 个常用 benchmark 函数<sup>[31]</sup>以及诺西肽分批发酵过程同步串联混合建模中 3 个黑箱模型网络参数的求取问题,将它与 CPSO<sup>[17]</sup>、QPSO<sup>[29]</sup>、TS<sup>[30]</sup>和 GA 四种算法进行寻优效果的对比.相关参数的设置情况为:benchmark 函数的维数包括 100 和 500 两种情况,其搜索范围如表 3 所示;黑箱模型的网络结构同第 4.2.2 节;CPSO、QPSO 和 TS 三种算法中,除了种群规模或候选集规模、最大迭代次数以外,其他参数的设置同原文献;GA 算法采用随机产生初始种群、单点交配、随机变异和轮盘赌方式选择并采用保优策略替换新种群中最差个体,其交配概率设置为 0.95,变异概率设置为 0.05;对于维数为 500 的 benchmark 函数,五种算法的种群规模或候选集规模设置为 200,最大迭代次数设置为 5000,其他情况则分别设置为 100 和 2000.每种优化算法均独立重复运行 20 次,其平均最优适应度函数值 (Mean best fitness function value, MBFFV) 和达到指定寻优精度所需迭代次数的中位值 (Median iteration requested, MIR) 如表 4~6 所示.

表 3 测试函数

Table 3 The benchmark functions

函数	表达式	搜索空间	最小值/最优位置
$f_1$	$\sum_{d=1}^D x_d^2$	$[-100, 100]^D$	0/(0, ..., 0)
$f_2$	$\sum_{d=1}^D (x_d^2 - 10 \cos(2\pi x_d) + 10)$	$[-5.12, 5.12]^D$	0/(0, ..., 0)
$f_3$	$\sum_{d=1}^D \frac{x_d^2}{4000} - \prod_{d=1}^D \cos\left(\frac{x_d}{\sqrt{d}}\right) + 1$	$[-600, 600]^D$	0/(0, ..., 0)

由表 4~6 可以看出,对于上述 3 个常用 benchmark 函数的极小化问题和网络参数求取问题,除了 100 维的 Sphere 函数以外,AMLPSO 算法的全局

搜索能力和收敛性能均优于其他四种优化算法, 这进一步验证了 AMLPSO 算法在求解高维复杂优化问题时的优良性能。

表 4 对于 3 个 benchmark 函数, 五种算法的 MBFFV  
Table 4 The MBFFV of AMLPSO, CPSO, QPSO, TS and GA on three benchmark functions

函数	维数	AMLPSO	CPSO	QPSO	TS	GA
$f_1$	100	3.77E-07	2.94E-06	<b>1.89E-07</b>	8.68E-05	9.53E-06
$f_1$	500	<b>1.39E-06</b>	1.11E-05	8.84E-05	9.21E-03	1.03E-03
$f_2$	100	<b>4.87E-01</b>	7.88E-01	9.26E-01	1.22E+02	9.69E+01
$f_2$	500	<b>7.15E+00</b>	1.43E+01	4.93E+01	1.55E+03	9.72E+02
$f_3$	100	<b>8.48E-06</b>	2.14E-05	1.53E-05	5.38E-03	9.17E-04
$f_3$	500	<b>1.29E-05</b>	2.91E-04	9.95E-05	3.69E-02	1.12E-02

表 5 对于 3 个 benchmark 函数, 五种算法的 MIR  
Table 5 The MIR of AMLPSO, CPSO, QPSO, TS and GA on three benchmark functions

函数	维数	精度	AMLPSO	CPSO	QPSO	TS	GA
$f_1$	100	8.0E-05	1452	1654	<b>1388</b>	1963	1875
$f_1$	500	1.0E-03	<b>2780</b>	3371	3778	$\infty$	4896
$f_2$	100	1.0E+00	<b>1193</b>	1584	1836	$\infty$	$\infty$
$f_2$	500	5.0E+01	<b>3354</b>	4062	4729	$\infty$	$\infty$
$f_3$	100	1.0E-03	<b>1336</b>	1476	1605	$\infty$	1883
$f_3$	500	5.0E-02	<b>3281</b>	3720	3979	4731	4484

表 6 对于网络参数求取问题, 五种算法的 MBFFV 和 MIR  
Table 6 The MBFFV and MIR of AMLPSO, CPSO, QPSO, TS and GA on the problem of determining network parameters

指标	精度	AMLPSO	CPSO	QPSO	TS	GA
MBFFV	-	<b>0.0127</b>	0.0333	0.0194	0.0897	0.0685
MIR	0.095	<b>1107</b>	1189	1272	1626	1469

## 5 结论

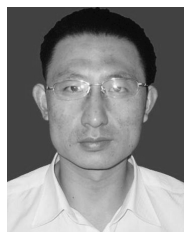
建立准确可靠的过程模型是实现发酵过程优化的基础和前提. 针对现有串联混合建模方法需要利用插值数据来构建中间变量黑箱模型的不足, 提出一种可将黑箱模型构建问题转化为动态模型参数辨识问题的同步串联混合建模方法, 并采用所提出的 AMLPSO 算法求取黑箱模型的网络参数, 利用均匀设计思想确定黑箱模型的网络结构. 通过诺西肽分批发酵过程实际生产数据验证了所提出方法的有效性, 结果表明: 同步串联混合建模方法所建过程模型的泛化性能优于现有串联混合方法, 可为过程优化奠定良好的模型基础; AMLPSO 算法通过引入多精

英学习策略和惯性权重自适应调整策略, 能有效避免算法陷入局部最优, 并获得较快的收敛速度. 本文所提出的建模方法对于解决其他类似工业过程的建模问题也有一定参考意义.

## References

- Shi Zhong-Ping, Pan Feng. *Fermentation Process Analysis, Control and Detection Technology (2nd Edition)*. Beijing: Chemical Industry Press, 2010. 1-10  
(史仲平, 潘丰. 发酵过程解析、控制与检测技术 (第 2 版). 北京: 化学工业出版社, 2010. 1-10)
- Sharma V, Mishra H N. Unstructured kinetic modeling of growth and lactic acid production by *Lactobacillus plantarum* NCDC 414 during fermentation of vegetable juices. *LWT-Food Science and Technology*, 2014, **59**(2): 1123-1128
- Wang R F, Koppram R, Olsson L, Franzén C J. Kinetic modeling of multi-feed simultaneous saccharification and co-fermentation of pretreated birch to ethanol. *Bioresource Technology*, 2014, **172**: 303-311
- Dragoi E N, Curteanu S, Galaction A I, Cascaval D. Optimization methodology based on neural networks and self-adaptive differential evolution algorithm applied to an aerobic fermentation process. *Applied Soft Computing*, 2013, **13**(1): 222-238
- Wang J L, Feng X Y, Yu T. A geometric approach to support vector regression and its application to fermentation process fast modeling. *Chinese Journal of Chemical Engineering*, 2012, **20**(4): 715-722
- Setoodeh P, Jahanmiri A, Eslamloueyan R. Hybrid neural modeling framework for simulation and optimization of diauxic-involved fed-batch fermentative succinate production. *Chemical Engineering Science*, 2012, **81**: 57-76
- Chen Jin-Dong, Pan Feng. Hybrid modeling for penicillin fermentation process. *CIESC J*, 2010, **61**(8): 2092-2096  
(陈进东, 潘丰. 青霉素发酵过程中的混合建模. 化工学报, 2010, **61**(8): 2092-2096)
- James S, Legge R, Budman H. Comparative study of black-box and hybrid estimation methods in fed-batch fermentation. *Journal of Process Control*, 2002, **12**(1): 113-121
- Wang X F, Chen J D, Liu C B, Pan F. Hybrid modeling of penicillin fermentation process based on least square support vector machine. *Chemical Engineering Research and Design*, 2010, **88**(4): 415-420
- Zorretto L F M, Filho R M, Wolf-Maciel M R. Processing development through artificial neural networks and hybrid models. *Computers and Chemical Engineering*, 2000, **24**(2-7): 1355-1360
- Laursen S Ö, Webb D, Ramirez W F. Dynamic hybrid neural network model of an industrial fed-batch fermentation process to produce foreign protein. *Computers and Chemical Engineering*, 2007, **31**(3): 163-170
- Saraceno A, Curcio S, Calabró V, Iorio G. A hybrid neural approach to model batch fermentation of "ricotta cheese whey" to ethanol. *Computers and Chemical Engineering*, 2010, **34**(10): 1590-1596

- 13 Beluhan D, Beluhan S. Hybrid modeling approach to on-line estimation of yeast biomass concentration in industrial bioreactor. *Biotechnology Letters*, 2000, **22**(8): 631–635
- 14 Birol G, Ündey C, Cinar A. A modular simulation package for fed-batch fermentation: penicillin production. *Computers and Chemical Engineering*, 2002, **26**(11): 1553–1565
- 15 Shukla R, Chand S, Srivastava A K. Batch kinetics and modeling of gibberellic acid production by *Gibberella fujikuroi*. *Enzyme and Microbial Technology*, 2005, **36**(4): 492–497
- 16 Kennedy J, Eberhart R C. Particle swarm optimization. In: Proceedings of the 1995 IEEE International Conference on Neural Networks. Perth, WA: IEEE, 1995. 1942–1948
- 17 Yang C H, Tsai S W, Chuang L Y, Yang C H. An improved particle swarm optimization with double-bottom chaotic maps for numerical optimization. *Applied Mathematics and Computation*, 2012, **219**(1): 260–279
- 18 Liu Gang, Lao Song-Yang, Yuan Can, Hou Lv-Lin, Tan Dong-Feng. OACRR-PSO algorithm for anti-ship missile path planning. *Acta Automatica Sinica*, 2012, **38**(9): 1528–1537  
(刘钢, 老松杨, 袁灿, 侯绿林, 谭东风. 反舰导弹航路规划的 OACRR-PSO 算法. 自动化学报, 2012, **38**(9): 1528–1537)
- 19 Li M S, Huang X Y, Liu H S, Liu B X, Wu Y, Xiong A H, Dong T W. Prediction of gas solubility in polymers by back propagation artificial neural network based on self-adaptive particle swarm optimization algorithm and chaos theory. *Fluid Phase Equilibria*, 2013, **356**(25): 11–17
- 20 Li Yong, Wu Min, Cao Wei-Hua, Lai Xu-Zhi, Wang Chun-Sheng. PSO-BP control algorithm of granulation process based on evaluation and optimization of granularity distribution. *Acta Automatica Sinica*, 2012, **38**(6): 1007–1016  
(李勇, 吴敏, 曹卫华, 赖旭芝, 王春生. 基于粒度分布评估与优化的制粒过程 PSO-BP 控制算法. 自动化学报, 2012, **38**(6): 1007–1016)
- 21 Sheng Yang, Lai Xu-Zhi, Wu Min. Position control of a planar three-link underactuated mechanical system based on model reduction. *Acta Automatica Sinica*, 2014, **40**(7): 1303–1310  
(盛洋, 赖旭芝, 吴敏. 基于模型降阶的平面三连杆欠驱动机械系统位置控制. 自动化学报, 2014, **40**(7): 1303–1310)
- 22 Jia D L, Zheng G X, Qu B Y, Khan M K. A hybrid particle swarm optimization algorithm for high-dimensional problems. *Computers and Industrial Engineering*, 2011, **61**(4): 1117–1122
- 23 Sedki A, Ouazar D. Hybrid particle swarm optimization and differential evolution for optimal design of water distribution systems. *Advanced Engineering Informatics*, 2012, **26**(3): 582–591
- 24 Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. *Neural Networks*, 1989, **2**(5): 359–366
- 25 Fang Kai-Tai, Ma Chang-Xing. *Orthogonal and Uniform Experimental Design*. Beijing: Science Press, 2001. 241–247  
(方开泰, 马长兴. 正交与均匀试验设计. 北京: 科学出版社, 2001. 241–247)
- 26 Benazet F, Cartier M, Florent J, Godard C, Jung G, Lunel J, Mancy D, Pascal C, Renaut J, Tarridec P, Theilleux J, Tissier R, Dubost M, Ninet L. Nosiheptide, a sulfur-containing peptide antibiotic isolated from *Streptomyces actuosus* 40037. *Experientia*, 1980, **36**(4): 414–416
- 27 Koutinas A A, Wang R, Kookos I K, Webb C. Kinetic parameters of *Aspergillus awamori* in submerged cultivations on whole wheat flour under oxygen limiting conditions. *Biochemical Engineering Journal*, 2003, **16**(1): 23–34
- 28 Khan N S, Mishra I M, Singh R P, Prasad B. Modeling the growth of *Corynebacterium glutamicum* under product inhibition in *L*-glutamic acid fermentation. *Biochemical Engineering Journal*, 2005, **25**(2): 173–178
- 29 Sun J, Fang W, Palade V, Wu X J, Xu W B. Quantum-behaved particle swarm optimization with Gaussian distributed local attractor point. *Applied Mathematics and Computation*, 2011, **218**(7): 3763–3775
- 30 Cvijović D, Klinowski J. Taboo Search: an approach to the multiple minima problem. *Science*, 1995, **267**(5198): 664–666
- 31 Yao X, Liu Y, Lin G M. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 1999, **3**(2): 82–102



杨强大 东北大学讲师. 主要研究方向为复杂工业过程建模与优化, 软测量技术. 本文通信作者.

E-mail: yangqd@smm.neu.edu.cn

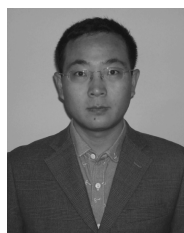
(YANG Qiang-Da Lecturer at Northeastern University. His research interest covers modeling and optimization of complex industrial processes, and soft sensing technology. Corresponding author of this paper.)



张卫军 东北大学教授. 主要研究方向为热工过程建模、优化与控制, 系统节能理论与应用.

E-mail: zhangwj@smm.neu.edu.cn

(ZHANG Wei-Jun Professor at Northeastern University. His research interest covers modeling, optimization and control of thermal processes, theory and application of system energy saving.)



牛大鹏 东北大学副教授. 主要研究方向为复杂工业过程建模与优化.

E-mail: niudapeng@ise.neu.edu.cn

(NIU Da-Peng Associate professor at Northeastern University. His research interest covers modeling and optimization of complex industrial processes.)