

多收率约束的催化裂化反再系统改进差分进化操作优化

曹 蕙, 陈庆达, 丁进良[†]

(东北大学 流程工业综合自动化国家重点实验室, 辽宁 沈阳 110819)

摘要: 针对现有催化裂化(FCC)装置操作优化中未根据市场需求考虑多产品收率约束的问题, 本文提出了一种求解多产品收率约束催化裂化反再系统操作优化的改进差分进化(iDE)算法。首先针对FCC操作优化中约束多和不同操作变量的可行范围差异大的特点, 设计了一种协同交互变异策略产生变异个体, 以提高算法的开发和探索能力; 其次提出了一种具有修复功能的参数自适应策略来更新变异因子和交叉因子。此外考虑到FCC操作优化具有时效强的特点, 提出了对每一代种群中最好个体实施加强搜索的方法, 以提高算法的收敛速度。仿真结果表明: 在求解多产品收率的FCC反再系统操作优化问题上, 该算法具有较强的全局寻优能力、鲁棒性以及较快的收敛速度。

关键词: 催化裂化; 进化算法; 操作优化; 差分进化; 协同变异; 加强搜索

引用格式: 曹蕙, 陈庆达, 丁进良. 多收率约束的催化裂化反再系统改进差分进化操作优化. 控制理论与应用, 2019, 36(8): 1207–1216

DOI: 10.7641/CTA.2018.80337

Operational optimization of fluid catalytic cracking reaction-regeneration system with multi-yield constraints using improved differential evolution algorithm

CAO Bei, CHEN Qing-da, DING Jin-liang[†]

(State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang Liaoning 110819, China)

Abstract: In view of existing operational optimization of fluid catalytic cracking (FCC) reaction-regeneration system has not considered multi-yield constraints of product demand, this paper proposes an improved differential evolution (iDE) algorithm to solve the operational optimization problem with multi-yield constraints in FCC unit. Firstly, considering there are many constraints in the FCC operational optimization, and the feasible ranges of different operating variables varies greatly, the cooperative interaction mutation strategy is designed to generate mutants, improving exploitation and exploration capabilities of the algorithm. Secondly, a parameter adaptation strategy with self-repairing capability is proposed to update crossover probability and scaling factor. Additionally, considering that the quick requirement of the operational optimization in FCC unit, an enhanced search strategy is presented to further exploit the best individual in each generation, enhancing the convergence speed of the iDE algorithm. The simulation experiments on solving the operational optimization with multi-yield constraints in FCC unit show that it is superior to the art-of-the-state algorithms in global optimization capability, robustness and convergence.

Key words: catalytic cracking; evolutionary algorithms; operational optimization; differential evolution; coordination mutation; enhanced search

Citation: CAO Bei, CHEN Qingda, DING Jinliang. Operational optimization of fluid catalytic cracking reaction-regeneration system with multi-yield constraints using improved differential evolution algorithm. *Control Theory & Applications*, 2019, 36(8): 1207–1216

1 引言

催化裂化(fluid catalytic cracking, FCC)是炼油厂中重要的二次加工过程, 其主要是在高温催化剂的作用下将来自常减压装置的减压蜡油等重质油裂解成

具有高附加值的轻质油, 如汽油、柴油等。当前约有50%的汽油来自于催化裂化装置^[1]。FCC反应再生系统(反再系统)在催化裂化过程中占据着核心地位, 其运行状况决定着产品的产量及性质^[2]。对该系统中操

收稿日期: 2018-05-08; 录用日期: 2018-10-24。

[†]通信作者。E-mail: jlding@mail.neu.edu.cn; Tel.: +86 24-83684245。

本文责任编辑: 李少远。

国家自然科学基金项目(61590922, 61525302, 61621004)资助。

Supported by the National Natural Science Foundation of China (61590922, 61525302, 61621004).

作变量进行小的改进都有可能带来较大的经济利润^[3]。因此,如何根据市场需求和装置的运行能力优化其操作条件对提高企业的经济效益具有重要的研究意义。

当市场对产品的需求量或生产计划发生变动时,FCC中当前的操作变量值可能改变了装置的最优运行工况,进而需要改变一个或多个操作变量值使装置运行在一个新最优工况下。然而由于催化裂化过程是一个高复杂度、强非线性的动态过程,很难通过定性或定量地分析操作变量的改变值对产品产量和性质的影响,进而难以给出一组最佳的操作条件^[1]。

为了能够确定FCC的最佳操作条件,学术界和工业界已经做出了巨大的贡献。在炼厂中,生产经营部门一般通过定期采用HYSYS等化工流程模拟软件对实际生产过程进行模拟,然后对影响产品收率的各变量进行敏感度分析,以便选取一组最佳操作变量值^[4],或采用模拟流程软件中的优化求解器对操作变量进行优化取值,这种方法虽简单易行,但软件中的优化求解器均是采用传统数学优化求解方法,如序列二次规划法^[4]、非线性规划法^[5]、改进序列二次规划法^[6-7]等直接对所建立的机理模型进行优化,其对目标函数和可行域的要求比较高,同时对实际生产过程中不确定信息的处理能力差。文献[8]基于后向传播算法(back propagation, BP)建立的汽油收率优化模型,采用遗传算法(genetic algorithm, GA)对其操作变量进行了优化。但他们仅研究了最大化汽油收率的问题,却没有考虑柴油等其他高附加值产品所带来的经济效益。文献[9]采用支持向量机以最大化轻质油收率之和为目标建立FCC的优化模型,然后采用粒子群优化算法(particle swarm optimization, PSO)算法进行优化求解。然而市场上对不同轻质油产品需求各异,仅考虑最大化汽油收率或轻质油收率是很难满足市场对其他产品的需求。差分进化算法因其简单易行、收敛性好等优点,被广泛应用到多智能体系统^[10-11]、高炉炼铁^[12]等各领域中。文献[13]采用传统差分进化(differential evolution algorithm, DE)算法以最大化FCC经济效益为目标,优化了反应再生系统的操作变量值,但该文仅采用了DE/rand/1/bin作为算法的变异策略,造成了算法的收敛速度慢,以及实际生产操作滞后,影响操作优化效果。

为了能够更好的优化FCC反再系统的操作条件设定值,最大化炼厂的经济效益。本文考虑了市场上各种产品油的需求量存在波动的特点,采用具备学习速度快、泛化性能好等优点的集成随机权神经网络^[14]建立了汽油、柴油、液化气的收率模型,以操作变量和满足市场需求的各产品收率的可行值范围为约束,以

最大化经济效益为目标建立了FCC反再系统操作变量的优化模型。针对多产品收率约束的催化裂化反再系统操作优化问题,本文提出了一种改进差分进化算法。首先针对多收率FCC操作优化中约束多和不同操作变量可行范围差异大的特点,提出了采用协同交互变异策略对个体进行变异;考虑到算法中交叉因子和变异因子对算法性能的影响较大,且FCC生产过程十分复杂很难事先确定该参数值的特点,提出了采用具有修复功能的参数自适应策略来确定算法中的交叉因子和变异因子;此外针对FCC中操作优化时效性强的特点,提出采用加强搜索策略对每一代种群中的最好解进行加强搜索,以提高算法的收敛速度,快速找到全局优解。最后针对催化裂化反应再生系统操作优化问题,将iDE算法与其他算法做了对比仿真分析。

2 问题描述

2.1 催化裂化反再过程简介

催化裂化反再装置的结构图如图1所示。来自常减压装置的减压蜡油经加热炉预热后,在反应器中提升管的底部与来自再生器中的高温催化剂接触,减压蜡油被瞬间气化生成反应油气。反应油气与催化剂颗粒在提升蒸汽的作用下沿着提升管上升,在上升过程中,反应油气在催化剂的作用下发生裂化反应得到烃化物和焦炭,焦炭会沉积在催化剂的表面使催化剂失去活性。当烃化物、失活的催化剂(待生催化剂)到达提升管的顶端后,在提升管出口的旋风分离器的作用下,烃化物、催化剂分离开来。烃化物经沉降器顶部的集气室进入分馏装置进行产品的分离,而待生催化剂则进入反应器的汽提段。催化剂在汽提段中与汽提蒸汽逆流接触以置换待生催化剂所携带的油气,置换后的待生催化剂经待生催化剂管线进入再生器的烧焦罐中,通过燃烧掉待生催化剂表面沉积的焦炭实现催化剂的再生。再生后的催化剂(再生催化剂)和燃烧过程产生的烟气在主风的作用下向上运动,最终经稀相管上部的分离器将催化剂和烟气分离。分离后的催化剂经再生催化剂管线进入反应器,实现催化剂的循环利用。烧焦产生的烟气通过再生器顶部的烟机等装置排出。

2.2 优化问题描述

在原料油和催化剂类型确定的前提下,影响产品性质和产量的主要因素是操作条件。在实际生产中,当反应温度不变时,提高原料油预热温度,则焦炭、汽油和液化气等产品的收率降低,柴油的收率增加;原料油流量越大,即单位时间内的装置加工量越大,总的产品产量也越大;提高反应温度则原料油的转化率提高;再生温度和再生压力均影响着再生催化剂的循环量;提升蒸汽流量为催化裂化反应提供流化床,其流量越大,则催化裂化的反应时间越短,而反应时间

的长短决定着催化裂化反应的深度, 反应深度不同则各产品的收率不同。即不同的操作变量对催化裂化生产过程中的产品分布均有着重要影响, 操作变量值不同, 各产品收率不同, 则炼厂的利润不同。

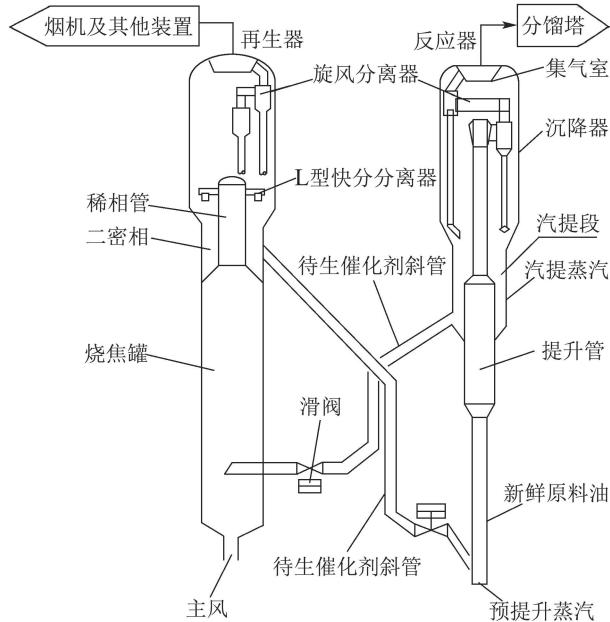


图1 催化裂化反再装置结构示意图

Fig. 1 Schematic diagram of FCC reaction-regeneration unit

汽油、柴油、液化气是FCC生产的主要利润来源。针对市场上对汽油等各产品的需求量存在波动的特点, 炼厂需要在各产品满足市场需求的前提下优化操作变量设定值, 以最大化炼厂的经济效益。

结合实际生产经验, 本文选择对产品性质和收率影响较大的原料油预热温度、原料油流量、反应温度、再生温度、再生压力、反应压力、提升蒸汽流量等7个操作条件作为操作变量。

2.3 优化模型

本文以最大化经济效益为目标, 以原料油预热温度、原料油流量、反应温度等7个变量为待优化的操作变量, 以各产品油的收率和操作变量的可行范围为约束, 建立了FCC反应再生系统操作优化模型。其中汽油、柴油、液化气等各产品收率模型是分别采用集成随机权神经网络, 以第2.2节所选取的7个操作变量为输入, 以汽油等各产品收率为输出建立的, 详细的集成随机权神经网络建模过程描述如附录1所示。本文的优化模型如下所示。

最大化炼厂利润=各产品销售总额-(原料油成本+原料油预热成本+提升蒸汽成本):

$$\begin{aligned} \max J = & \\ & \sum_{m=1}^3 y_m \times X_{.,2} \times T \times w_{\text{pro},m} - [X_{.,2} \times T \times w_{\text{oil}} + \\ & (X_{.,1} - t_s) \times X_{.,2} \times T \times w_{\text{ene}} + \end{aligned}$$

$$X_{.,7} \times T \times w_{\text{vap}}]. \quad (1)$$

装置生产能力约束:

$$X_{.,j}^{\min} \leq X_{.,j} \leq X_{.,j}^{\max}, j = 1, 2, \dots, D. \quad (2)$$

各产品收率约束:

$$y_m^{\min} \leq y_m \leq y_m^{\max}, m = 1, 2, 3, \quad (3)$$

其中各产品收率模型为

$$y_m = f_m(X_{.,1}, X_{.,2}, \dots, X_{.,D}), \quad (4)$$

m 为产品指数, m 取1, 2, 3, 分别代表汽油、柴油、液化气, 其中 y_m 表示第 m 种产品油的收率; y_m^{\max} 和 y_m^{\min} 分别为第 m 种产品油收率取值范围的上下限, 函数 $f_m(\cdot)$ 表示第 m 种产品油的收率预报模型; T 为FCC装置的加工时间; $w_{\text{pro},m}$ 代表第 m 种产品油的单价; w_{oil} 为原料油的单价; w_{vap} 为提升蒸汽的单价; t_s 为原料油的初始温度; w_{ene} 为进料油预热时其温度每升高1°的能耗成本; $X_{.,j}$ 表示任意一个解的第 j 维变量, 即第 j 个决策变量, j 取1, 2, ..., D , 依次代表原料油预热温度、原料油流量、反应温度、再生温度、再生压力、反应压力和提升蒸汽流量等决策变量, $X_{.,j}^{\max}$ 和 $X_{.,j}^{\min}$ 分别为任意一个解的第 j 维变量取值范围的上下限。

3 基本差分进化算法

基本的DE算法首先要初始化一个规模为 n 的种群 $X_i (i=1, 2, \dots, n)$, $X_i \in [X_i^{\min}, X_i^{\max}]$, 且 X_i 为 D 维向量, $X_i^{\max} = [X_{i,1}^{\max} \dots X_{i,D}^{\max}]$, 同理 $X_i^{\min} = [X_{i,1}^{\min} \dots X_{i,D}^{\min}]$; 然后对种群中的每个个体进行不断的变异, 交叉, 选择操作。

第 g 代第 i 个个体 $X_{i,g}$ 经变异可产生变异个体 $V_{i,g}$, 常用的变异策略^[15]有

DE/rand/1/bin:

$$V_{i,g} = X_{r1,g} + F \cdot (X_{r2,g} - X_{r3,g}). \quad (5)$$

DE/best/1/bin:

$$V_{i,g} = X_{\text{best},g} + F \cdot (X_{r2,g} - X_{r3,g}). \quad (6)$$

DE/current-best/1/bin:

$$\begin{aligned} V_{i,g} = X_{i,g} + F \cdot (X_{\text{best},g} - X_{i,g}) + \\ F \cdot (X_{r2,g} - X_{r3,g}). \end{aligned} \quad (7)$$

式(5)–(7)中 r_1, r_2, r_3 分别是 $[0, n]$ 范围内的一个随机整数, 且 $r_1 \neq r_2 \neq r_3 \neq i$, F 为变异因子, $X_{\text{best},g}$ 为第 g 代种群中最好的个体。

变异个体 $V_{i,g}$ 和个体 $X_{i,g}$ 进行交叉操作, 可得到试验个体 $U_{i,g}$, 其常用的交叉策略^[15]如下所示:

$$U_{i,j,g} = \begin{cases} V_{i,j,g}, & CR_i < \text{rand} \& j = j_{\text{rand}}, \\ X_{i,j,g}, & \text{其他}. \end{cases} \quad (8)$$

式(8)中: CR_i 为第 i 个个体所对应的交叉因子, rand

表示在 $[0, 1]$ 范围内随机产生一个实数, j_{rand} 为 $[1, D]$ 中的一个随机整数.

根据适应度值的大小从当前代中的试验个体 $U_{i,g}$ 和个体 $X_{i,g}$ 中挑选较好的个体进入下一代, 其选择操作^[15]如下所示:

$$X_{i,g+1} = \begin{cases} U_{i,g}, & \text{fitness}(U_{i,g}) \geq \text{fitness}(X_{i,g}), \\ X_{i,g}, & \text{其他.} \end{cases} \quad (9)$$

式(9)中 $\text{fitness}(\cdot)$ 为适应度函数值.

4 改进差分进化算法

4.1 协同交互变异策略

FCC反再系统中各操作变量的可行范围不同且差异较大, 这使得解空间有较强的不规则性, 其多产品的收率约束使得操作优化过程中可行域较小, 因而如何在FCC操作优化不规则的解空间中快速找到全局最好解具有重要意义.

DE/rand/1/bin变异随机性大, 全局搜索能力强, 但收敛速度慢. 而DE/current-best/1/bin变异收敛速度快, 但易陷入局部最优, 出现早熟. 因此, 本文使各个体采用这两种变异机制进行协同交互变异, 以便iDE算法快速找到全局最好解, 使算法具备较强的全局寻优能力.

在算法的变异阶段通过引入一个交互周期因子 S 来控制DE/rand/1/bin和DE/current-best/1/bin变异机制的协同交互, 进而平衡差分进化算法探索和开发能力. 具体做法是用 $\text{Count } 1_i$ 记录种群中个体 $X_{i,g}$ 采用DE/rand/1/bin变异连续不更新的次数, $\text{Count } 2_i$ 记录种群中个体 $X_{i,g}$ 采用该DE/current-best/1/bin变异连续不更新的次数. 若个体 $X_{i,g}$ 采用DE/rand/1/bin变异连续 S 次迭代不更新, 即 $\text{Count } 1_i = S$ 时, 则使该个体 $X_{i,g}$ 采用DE/current-best/1/bin变异, 若个体 $X_{i,g}$ 采用该变异后也出现连续 S 次迭代不更新, 即 $\text{Count } 2_i = S$ 时, 则使个体 $X_{i,g}$ 再次用DE/rand/1/bin变异, 采用该种协同交互方法, 可以很好的平衡算法的探索和开发能力, 进而提高算法的全局寻优能力.

4.2 自修复参数自适应策略

研究表明交叉因子 CR 和变异因子 F 对DE算法的性能有着重要影响^[16-17]. FCC生产过程十分复杂, 很难在其操作优化时事先选择一组很合适的变异因子和交叉因子参数值, 因此本文提出了一种具有修复功能的参数自适应策略, 以改善算法的寻优性能和鲁棒性, 具体策略如下:

F_i 的生成方式同具有可选外部存档的自适应差分进化算法^[18](adaptive differential evolution with optional external archive, JADE)一样, 即每代中不同个体 X_i 的变异因子 F_i 都独立产生, 且服从柯西分布.

协同交互变异算法如下:

Input: $X_{i,g}$, S , Count 1_i , Count 2_i

Output: $V_{i,g}$

Step 1 变异机制选择.

if Count $1_i = S$

$Fg_i = 1$; Count $1_i = 0$;

end if

if Count $2_i = S$

$Fg_i = 0$; Count $2_i = 0$;

end if

Step 2 变异机制分配.

if $Fg_i = 0$

 个体 $X_{i,g}$ 按式(5)进行变异, 产生 $V_{i,g}$;

end if

if $Fg_i = 1$

 个体 $X_{i,g}$ 按式(7)进行变异, 产生 $V_{i,g}$;

end if

即

$$F_i = \text{randc}_i(F_m, 0.1), \quad (10)$$

$\text{randc}_i(\cdot)$ 表示为个体 X_i 随机产生一个服从柯西分布的变异因子 F_i , F_m 的初始值为0.5, 其在每一代结束时更新, 更新方式如下:

$$F_m = (1 - c) \cdot F_m + c \cdot \text{mean}_L(S_F). \quad (11)$$

式(11)中 c 为 $[0, 1]$ 区间中的一个正整数, 其中: S_F 为记录第 g 代中成功变异的个体所对应的 F_i 的集合; $\text{mean}_L(\cdot)$ 为Lehmer均值函数, 其计算方式如下:

$$\text{mean}_L(S_F) = \frac{\sum_{F \in S_F} F^2}{\sum_{F \in S_F} F}. \quad (12)$$

F_i 越大进化速度越快, 但 F_i 太大则不利于算法局部搜索; F_i 越小则局部搜索更细致, 但却降低了算法的全局搜索能力. 因此为了平衡种群的全局搜索和局部搜索能力, 本文进行如下参数自修复操作: 将 F_i 的取值限制在 $[0, 1]$ ^[17]内. 即当 $F_i > 1$, 则利用种群中最好解和当前个体的信息来修复 F_i 的取值, 即

$$F_i = \frac{\text{fitness}(X_{\text{best},g})}{\text{fitness}(X_{\text{best},g}) + \text{fitness}(X_{i,g})}. \quad (13)$$

式(13)保证了 F_i 在 $[0.5, 1]$ 的范围取值, 保留了 F_i 取较大值的倾向. 如果 $F_i \leq 0$, 则利用下式:

$$F_i = \frac{\text{fitness}(X_{i,g})}{\text{fitness}(X_{i,g}) + \text{fitness}(X_{\text{best},g})}, \quad (14)$$

可保证 F_i 在 $[0, 0.5]$ 范围内取值, 保留了 F_i 取较小值的倾向.

同理, 每代中不同个体 X_i 的交叉因子 CR_i 都独立

产生, 其取值也同JADE算法一样服从正态分布, 且 $CR_i \in [0, 1]$, 即

$$CR_i = \text{randn}_i(CR_m, 0.1), \quad (15)$$

其中 $\text{randn}_i(\cdot)$ 表示为个体 X_i 随机产生一个服从正态分布的交叉因子 CR_i , CR_m 的初始值为 0.5, 并在每一代结束的时候更新, 更新方式如下:

$$CR_m = (1 - c) \cdot CR_m + c \cdot \text{mean}_A(S_{\text{CR}}). \quad (16)$$

式(16)中: S_{CR} 为成功交叉个体所对应的交叉因子 CR_i 的集合, $\text{mean}_A(\cdot)$ 为算数平均值函数. 为了保证交叉因子在 $[0, 1]$ 范围内, 本文也采用了类似参数的自修复操作: 即当 $CR_i > 1$ 时,

$$CR_i = \frac{\text{fitness}(X_{\text{best},g})}{\text{fitness}(X_{\text{best},g}) + \text{fitness}(X_{i,g})}; \quad (17)$$

当 $CR_i \leq 0$ 时,

$$CR_i = \frac{\text{fitness}(X_{i,g})}{\text{fitness}(X_{i,g}) + \text{fitness}(X_{\text{best},g})}. \quad (18)$$

引入该自修复操作的参数自适应策略不仅高效的更新了 F_i 和 CR_i , 而且增加了种群的多样性、改善了算法的寻优性能、提高了算法的鲁棒性.

4.3 加强搜索策略

FCC反再系统操作优化的时效性强, 即需要在较短的时间内获得优化结果, 这使得iDE算法需在保证全局寻优能力的同时必须兼具较快的收敛速度. 贪婪变异机制DE/best/1/bin收敛速度快, 局部搜索能力强, 而且较好解具有好的搜索信息. 因此本文采用DE/best/1/bin对优化过程中每代协同变异后种群 X_{g+1} 中的最好个体 $X_{i,g+1}$ 进行一次加强搜索, 即再进行一次变异, 交叉得到 $X_{i,g+1}^*$, 如果 $X_{i,g+1}^*$ 经汽油、柴油、液化气收率模型计算出的各收率 $y_{1,i}, y_{2,i}, y_{3,i}$ 满足相应的约束条件, 且 $X_{i,g+1}^*$ 的适应度值比 $X_{i,g+1}$ 优秀时, 则用 $X_{i,g+1}^*$ 更新种群中的最好个体 $X_{i,g+1}$ 及全局最好解 $X_{\text{best},g+1}$, 反之, 则不更新. 该方法充分利用当前种群的最好解和其他个体信息引导种群快速向全局最优解靠近, 提高了iDE算法的收敛速度. 其加强搜索过程中的变异因子和交叉因子仍用协同交互变异后的种群中最好个体所对应的变异因子 F_i 和交叉因子 CR_i . 操作优化的解一般靠近边界, 因此本文对超越解空间边界的个体做以下边界处理^[19], 即

$$V_{i,j,g} = \begin{cases} (X_{i,j,g}^* + X_{i,j}^{\min}) / 2, & V_{i,j,g} < X_{i,j}^{\min}, \\ (X_{i,j,g}^* + X_{i,j}^{\max}) / 2, & V_{i,j,g} > X_{i,j}^{\max}. \end{cases} \quad (19)$$

这种方法在处理最好解在解空间边界附近问题上十分有效.

本文约束处理的方法是在初始化时利用种群初始化的随机性确保初始种群中的个体均为可行解, 若在进化过程中出现不可行解, 则使相应原始可行个体 $X_{i,g}$ 再次进行变异、交叉等操作重新产生新个体, 若连续两次重新进行变异、交叉等操作后产生的新个体仍为不可行解, 则令该新个体取原始可行个体值.

加强搜索算法具体如下:

Input: F_i, CR_i, X_{g+1} .

Output: X_{g+1} . 变异机制选择.

随机从当前中群中选取两个个体 X_{r1} 和 X_{r2} , 且 $X_{r1} \neq X_{i,g+1}, X_{r2} \neq X_{i,g+1}$;

Step 1 个体按式(6)进行变异产生 $X_{i,g+1}$;

Step 2 边界处理.

for $j = 1 : D$

$X_{i,j,g+1}^*$ 按式(19)进行处理;

end for

Step 3 交叉操作.

for $j = 1 : D$

$X_{i,j,g+1}^*$ 按式(8)进行交叉操作;

end for

Step 4 收率约束处理.

判断变异个体 $X_{i,g+1}^*$ 预报的汽油、柴油、液化气各收率 $y_{1,i}, y_{2,i}, y_{3,i}$ 是否均满足相应约束条件, 若不满足, 则跳回Step 1. 若连续两次重新变异后个体 $X_{i,g+1}^*$ 预报的收率仍不满足约束条件, 则不再跳回Step 1, 直接令 $X_{i,g+1}^* = X_{i,g+1}$, 即个体 $X_{i,g+1}$ 不变异, $X_{\text{best},g+1}$ 不更新;

Step 5 选择操作.

按式(9)更新个体 $X_{i,g+1}$ 和 $X_{\text{best},g+1}$.

4.4 算法流程

结合上述改进策略, iDE算法流程如图2所示.

5 仿真实验

5.1 仿真设置

市场上催化裂化原料油及其产品的价格是不断变化的, 因此当其价格不同时, 最大化经济效益时的目标函数就不同, 进而导致操作条件设定值的优化结果不同. 为了验证所提算法的鲁棒性, 本文结合2018年3月份某炼厂催化裂化原料油及其产品价格表, 选取了一段时间内的10组价格值如表1所示, 基于集成随机权神经网络模型分别构建了10个最大化催化裂化经济效益的测试案例.

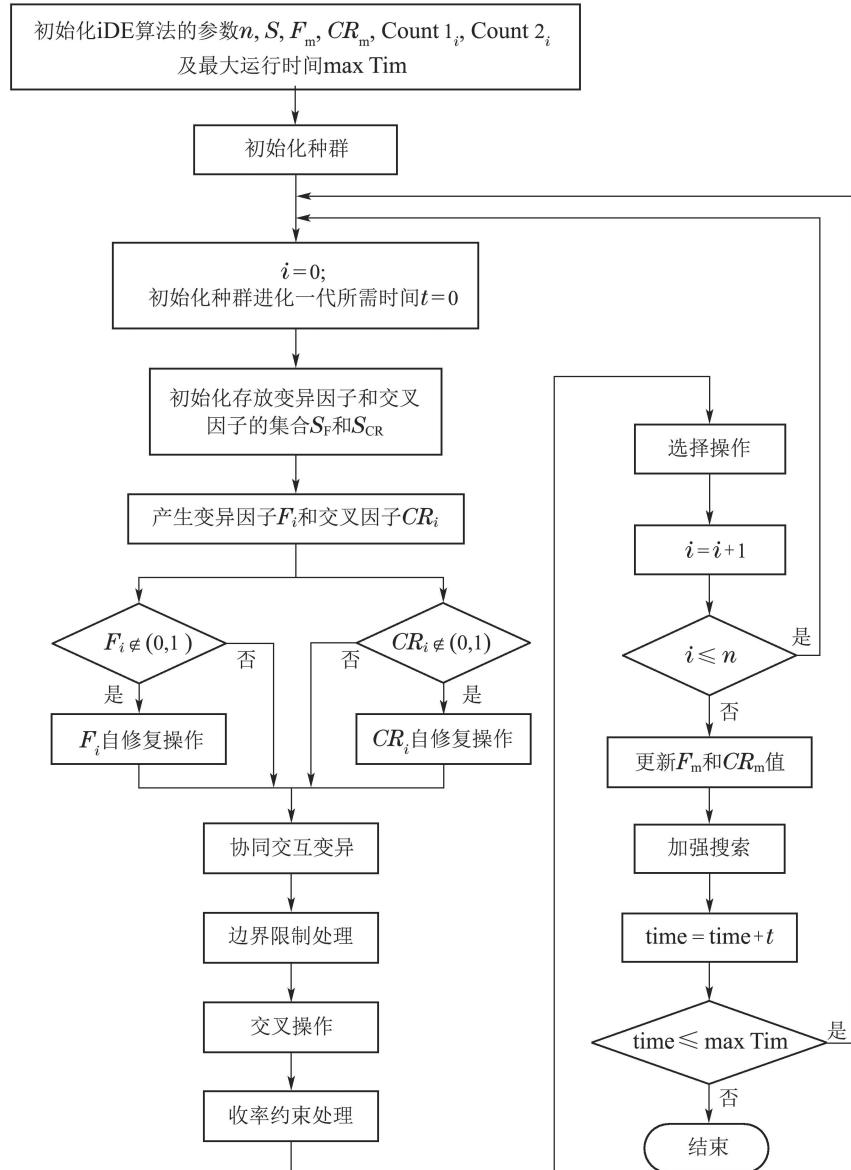


图2 iDE算法流程
Fig. 2 The flow of iDE algorithm

表1 催化裂化原料油及产品价格
Table 1 The price of FCC feedstock oil and each product

序号	汽油/(元·t ⁻¹)	柴油/(元·t ⁻¹)	液化气/(元·t ⁻¹)	原料油/(元·t ⁻¹)
1	7260	6370	3750	3550
2	7010	6540	3850	4040
3	7100	6260	3670	3870
4	7270	6390	3900	3960
5	7350	6510	3080	3600
6	6940	6570	3550	3970
7	7350	6580	3810	3880
8	6610	6610	3400	4120
9	6850	6490	3600	4350
10	6370	6890	3650	3770

该部分选取了具有代表性的差分进化算法: JADE 和自适应控制参数差分进化算法^[20](self-adapting control parameters differential evolution, jDE), 以及目前已在催化裂化相关优化问题上应用的传统的DE^[12], GA^[8]和PSO^[9]算法与本文所提iDE算法进行对比仿真分析。其中JADE, jDE, DE, GA和PSO算法的参数设置和原文保持一致, 为了公平起见, 本文算法的参数 $c=0.1$, $S=2$, $n=50$ 以及初始的 $CR_m=0.5$, $F_m=0.5$ 均保持不变。由于JADE, jDE, GA, PSO和传统的DE算法本身均不带约束处理, 而本文的优化问题需要对约束条件进行处理, 故对各算法均使用了第4.4节相同的约束处理方法。在所有仿真中, 种群中个体维度, 即决策变量个数 $D=7$, 进化的终止条件均为最大运行时间maxTim。由于实际生产十分注重效率, 即需要在较短的时间内作出优化决策, 因此

结合实际生产需求,本文取 $\text{maxTim} = 200$ s。针对不同的测试案例,各算法均独立运行20次,并计算其独立运行20次的全局最好解的平均值,其结果如表2所示。为了检验iDE算法的整体性能,本文在显著水平为0.05时对其进行t检验,测试结果如表3所示。

本文用“+,-,”分别表示iDE算法优于、持平于、差于被比较算法。此外,本文任意选取了一个测试案例,并给出了该案列下所有算法的收敛过程曲线,以便直观的观察和对比各算法的收敛性能和全局寻优能力,如图3所示。

表2 不同测试案例运行20次结果平均值数据及分析
Table 2 20 results average data and analysis of different test cases

测试案例	iDE	JADE	jDE	DE	GA	PSO					
1	1.11E+08	-	1.12E+08	+	1.07E+08	+	1.03E+08	+	1.07E+08		
2	1.09E+08	+	1.06E+08	+	1.06E+08	+	9.87E+07	+	1.01E+08	+	1.05E+08
3	1.09E+08	+	1.07E+08	+	1.05E+08	+	1.03E+08	+	1.00E+08	+	1.05E+08
4	1.12E+08	+	1.03E+08	+	1.03E+08	+	1.01E+08	+	9.86E+07	+	1.03E+08
5	1.09E+08	+	1.07E+08	=	1.09E+08	+	1.05E+08	+	1.02E+08	+	1.06E+08
6	1.08E+08	+	1.04E+08	+	1.03E+08	+	1.02E+08	+	9.95E+07	+	1.04E+08
7	1.12E+08	+	1.10E+08	+	1.08E+08	+	1.08E+08	+	1.05E+08	+	1.09E+08
8	1.02E+08	+	1.00E+08	+	1.00E+08	-	1.04E+08	+	9.58E+07	+	9.96E+07
9	1.05E+08	+	1.04E+08	+	1.03E+08	+	1.01E+08	+	9.85E+07	+	1.02E+08
10	1.02E+08	+	1.01E+08	+	1.01E+08	+	9.78E+07	+	9.49E+07	+	9.91E+07
+/-	+/-	9/0/1	9/1/0	9/0/1	10/0/0	10/0/0					

表3 不同测试案例上各算法检验结果
Table 3 T test results for each algorithms in different test cases

测试案例	JADE	jDE	DE	GA	PSO
1	4.25E-06	2.86E-06	5.34E-12	8.98E-14	9.84E-10
2	1.39E-02	9.00E-04	1.51E-15	3.81E-15	2.76E-08
3	1.82E-02	9.90E-06	6.04E-12	1.72E-15	1.24E-09
4	1.30E-03	2.19E-05	8.93E-11	1.23E-12	3.32E-08
5	8.42E-03	9.89E-07	4.41E-14	4.40E-15	1.32E-11
6	3.85E-02	1.31E-07	2.27E-16	1.25E-16	1.14E-10
7	2.23E-02	6.90E-05	3.33E-10	1.85E-13	1.34E-08
8	2.96E-03	2.22E-04	1.58E-04	8.41E-14	1.71E-09
9	2.26E-03	2.43E-04	1.16E-10	9.49E-13	1.40E-07
10	1.75E-04	1.23E-02	3.20E-10	2.62E-14	3.25E-07

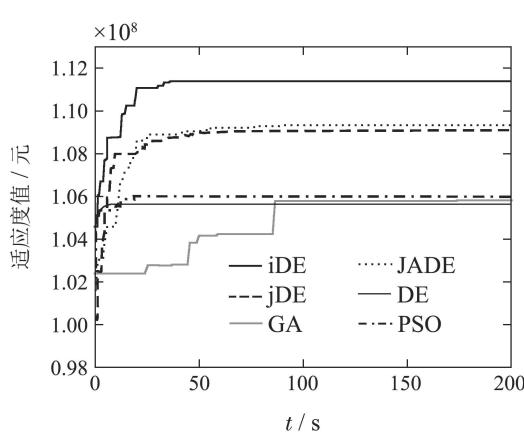


图3 各算法收敛过程曲线图

Fig. 3 The curve diagram of the convergence process of each algorithm

本文所有仿真均在处理器为Intel(R)Core(TM)i7-4770CPU@3.40 GHz的DELL PC机上进行,编程环境是MATLAB R2016b。

5.2 仿真结果及分析

从表2中的结果可以看出,在10个不同的测试案例上,本文提出的iDE算法的平均最好解均比GA和PSO算法好,与JADE和DE算法相比,则均各在9个案例上的平均最好解较好,而各有1个案例上的平均最好解较差;与传统DE算法相比,在9个不同案例上的平均最好解较好,有1个案例上其平均最好解持平。从表3的结果可知,各对比算法在显著水平为0.05下均与本文所提iDE算法的差异性较大,这说明表2的结果在不同测试案例上均具有统计学意义,即iDE算法的全

局寻优能力更好,这验证了协同交互变异策略能够很好的平衡开发和探索能力,使iDE算法避免早熟,找到全局最好解。此外,算法的性能好坏与其参数的控制紧密相关,因此在iDE算法在不同案例上的良好表现,也得益于本文具有修复功能的参数自适应策略对算法参数较好的控制。图3中各算法的收敛过程曲线可知,在运行时间为38 s左右时,iDE算法就已收敛。传统的DE和PSO算法虽比iDE算法收敛的更快,但它们显然陷入了局部最优,而JADE和jDE算法的收敛速度和最好解均不及iDE算法。GA算法的收敛速度最慢。这说明了对每代中的最好个体进行加强搜索可以使iDE算法更快的找到全局最好解,即加快收敛速度。

6 结论

针对多收率约束的催化裂化反应再生系统操作优化问题,提出的iDE算法通过集成DE/rand/1/bin和DE/current-best/1/bin的优点,对个体进行变异,这使得该算法具有较强的全局寻优能力;具有修复功能的交叉因子和变异因子自适应更新策略,使得算法的变异效率得到了有效的提升,鲁棒性得到了加强;此外,提出的对每代种群中最好解进行加强搜索的策略大大地加快了算法的收敛速度。综上所述,所提iDE算法在原料油和催化剂性质确定的条件下,有效地优化了操作变量的设定值,提高了经济效益。通过对不同优化案例仿真验证了iDE算法在求解多收率约束的催化裂化反应再生系统操作优化问题上具有很大的优势。

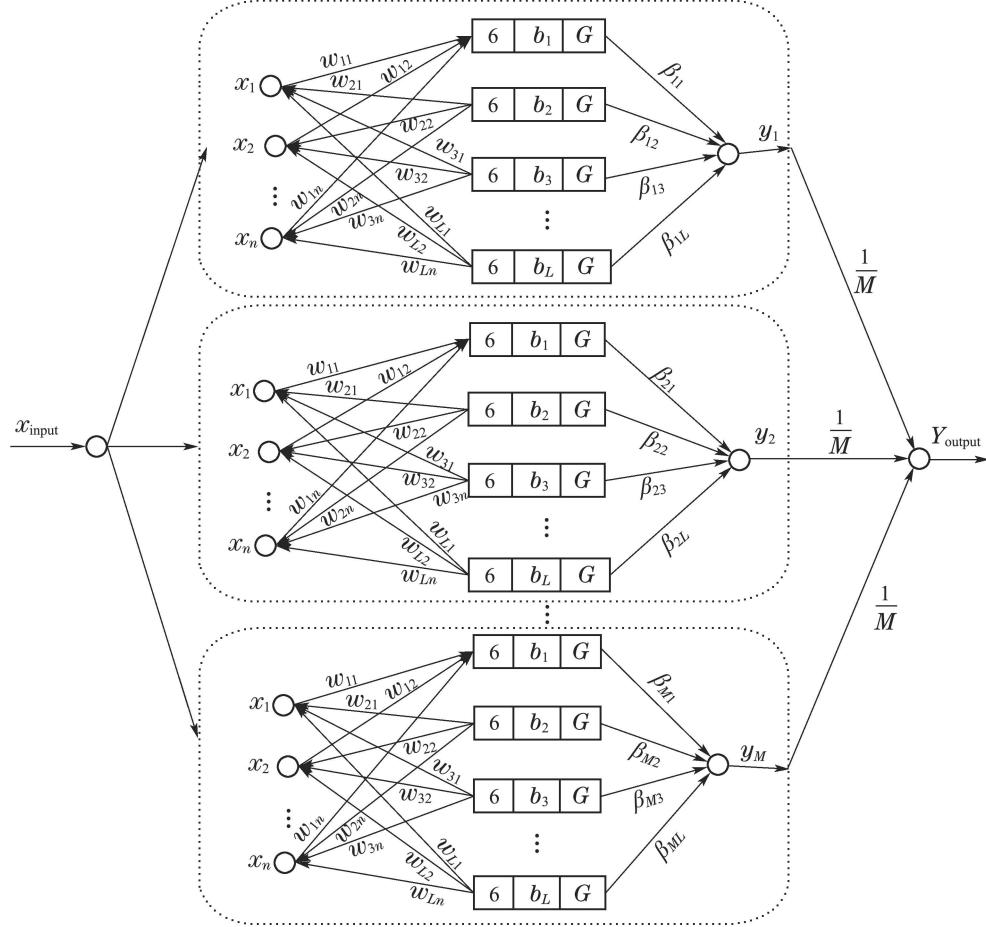
参考文献:

- [1] JOHN Y M, PATEL R, MUJTABA I M. Maximization of gasoline in an industrial FCC unit. *Energy and Fuels*, 2017, 31(5): 5645 – 5661.
- [2] LV Cuiying, WU Jianghong, LIN Huimin. Operation optimization of neural networks for reaction-reproduction systems of fluid catalytic cracking unit. *Computers and Applied Chemistry*, 2002, 19(4): 447 – 450.
(吕翠英, 吴江宏, 林慧敏. 催化裂化反再系统神经网络操作优化. 计算机与应用化学, 2002, 19(4): 447 – 450.)
- [3] HUANG Dexian, JIANG Yongheng, JIN Yihui. Present researchsituation, major bottlenecks, and prospect of refineryindustry process control. *Acta Automatica Sinica*, 2017, 43(6): 902 – 916.
(黄德先, 江永亨, 金以慧. 炼油工业过程控制的研究现状、问题与展望. 自动化学报, 2017, 43(6): 902 – 916.)
- [4] PINHEIRO C I C, FEMANDES J L, DOMINGUES L, et al. Fluid catalytic cracking(FCC)process modeling, simulation, and control. *Industrial and Engineering Chemistry Research*, 2012, 51(1): 1 – 29.
- [5] ALKAABNEH F, ALKAWALEE N, LLUBEGA W. Maximization of conversion in fluid catalytic cracker. *Proceeding of 2013 International Conference on Industrial Engineering and Systems Management*. Rabat: IEEE, 2013, 10: 1 – 5.
- [6] HAN I S, RIGGS J B, CHUNG C B. Modeling and optimization of a fluidized catalytic cracking process under full and partial combustion modes. *Chemical Engineering and Processing Process Intensification*, 2004, 43(8): 1063 – 1084.
- [7] EILL R C, XUAN L I, RIGGS J B. Modeling and optimization of a model IV fluidized catalytic cracking unit. *AIChE*, 1998, 44(9): 2068 – 2079.
- [8] OUYANG Fusheng, FANG Weigang, TANG Jiarui, et al. Fluid catalytic cracking MIP process BP neural network genetic algorithm. *Petroleum Processing and Petrochemicals*, 2016, 47(5): 95 – 100.
(欧阳福生, 方伟刚, 唐嘉瑞, 等. 以BP神经网络为基础MIP工艺过程产品分布优化. 石油炼制与化工, 2012, 43(5): 76 – 81.)
- [9] SUN Zhongchao, SHAN Honghong, LIU Yibin, et al. Modeling and optimization for catalytic cracking products of heavy oil based on support vector regression. *Petroleum Processing and Petrochemicals*, 2012, 43(5): 76 – 81.
(孙忠超, 山红红, 刘熠斌, 等. 基于支持向量回归的重油催化裂化产物建模及优化. 石油炼制与化工, 2012, 43(5): 76 – 81.)
- [10] ZHOU Yanping, GU Xingsheng. Development of differential evolution algorithm. *Control and Instruments in Chemical Industry*, 2007, 34(3): 1 – 5.
(周艳平, 顾幸生. 差分进化算法研究进展. 化工自动化及仪表, 2007, 34(3): 1 – 5.)
- [11] GUI Haixia, ZHANG Guofu, SU Zhaopin, et al. An algorithm for overlapping coalition structure generation based on differential evolution and encoding repairs. *Control Theory & Applications*, 2018, 35(2): 215 – 223.
(桂海霞, 张国富, 苏兆品, 等. 一种基于差分进化和编码修正的重叠联盟结构生成算法. 控制理论与应用, 2018, 35(2): 215 – 223.)
- [12] JIANG Zhaojun, YIN Juping, GUI Weihua, et al. Prediction for blast furnace silicon content in hot metal based on composite differential evolution algorithm and extreme learning machine. *Control Theory & Applications*, 2016, 35(8): 1089 – 1095.
(蒋朝辉, 尹菊萍, 桂卫华, 等. 基于复合差分进化算法与极限学习机的高炉铁水硅含量预报. 控制理论与应用, 2016, 35(8): 1089 – 1095.)
- [13] LONG J, MAO M S, ZHAO G Y. Model optimization for an industrial fluid catalytic cracking riser-regenerator unit by differential evolution algorithm. *Liquid Fuels Technology*, 2015, 33(13): 1380 – 1387.
- [14] DING J, WANG H, LI C, et al. An online learning neural network ensembles with random weights for regression of sequential data stream. *Soft Computing*, 2016, 21(20): 1 – 19.
- [15] QIN A K, HUANG V L, SUGANTHAN P N. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Press*, 2009, 13(2): 398 – 417.
- [16] ZAHARIE D. On the explorative power of differential evolutionary-algorithm. *Verhandlungen Der Deutschen Gesellschaft Fur Innere Medizin*, 2001, 39(6): 551 – 554.
- [17] DASGUPTA S, DAS S, BRISWAS A, et al. On stability and convergence of the population dynamics in differential evolution. *AI Commun*, 2009, 22(1): 1 – 20.
- [18] ZHANG J, SABDERSON A C. JADE: Adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, 2009, 13(5): 945 – 958.
- [19] PRICE K V, STOM R M, LAMPINEN J A. *Differential Evolution: A Practical Approach To Global Optimization*. New York: Springer Verlag, 2005.
- [20] BREST J, GREINER S, BOSKOVIC B, et al. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*, 2006, 10(6): 646 – 657.

附录

A1) 集成随机权神经网络.

集成随机权神经网络是由多个相同的单隐层前馈神经网络集合而成,其结构如图A1所示。



图A1 集成随机权神经网络结构图

Fig. A1 The structure chart of ensemble neural networks with random weights

各个单隐层前馈个体网络的输入 X_{input} 相同, 输出 y 均表示为

$$y(X_{\text{input}}^N) = \sum_{j=1}^L \beta_j G(w_j^T \cdot X_{\text{input}}^N + b_j). \quad (\text{A1})$$

式中: $X_{\text{input}}^N = [x_1^N \ x_2^N \ \cdots \ x_D^N]$ 表示第 N 组样本数据, D 表示样本数据的维度; L 表示隐藏层节点数; w_j 表示输入权值, 其在 $[-1, 1]$ 范围内随机产生; $b_j \in \mathbb{R}$ 表示隐藏层节点阀值; $G(\cdot)$ 表示隐藏层节点激励函数, 其常采用sigmoid函数; $\beta = [\beta_1 \ \beta_2 \ \cdots \ \beta_L]$ 为输出权值. 集成随机权神经网络的输出是 M 个单隐层前馈个体网络输出的平均加权, 即

$$Y_{\text{output}} = \frac{1}{M} \sum_{m=1}^M y_m(X_{\text{input}}). \quad (\text{A2})$$

式中 M 为集成规模, 表示第 m 个个体网络.

所有个体网络的输出权值构成全局输出权值矩阵 $\beta_{\text{ens}} = [\beta_{11} \ \beta_{12} \ \cdots \ \beta_{1L} \ \cdots \ \beta_{M1} \ \beta_{M2} \ \cdots \ \beta_{ML}]_{ML \times 1}$. 集成随机权网络模型的关键就是确定全局输出权值矩阵.

由文献[14]可知, 隐藏层到输出层可以表示成如下矩阵形式:

$$H_{\text{corr}} \beta_{\text{ens}} = T_h, \quad (\text{A3})$$

其中: H_{corr} 称为隐藏层相关矩阵, T_h 称为目标矩阵, 即

$$H_{\text{corr}} = \begin{bmatrix} C_1 H_1^T H_1 + 2\alpha_1 & C_2 H_1^T H_2 & \cdots & C_2 H_1^T H_M \\ C_2 H_2^T H_1 & C_1 H_2^T H_2 + 2\alpha_2 & \cdots & C_2 H_2^T H_M \\ \vdots & \vdots & \ddots & \vdots \\ C_2 H_M^T H_1 & C_2 H_M^T H_2 & \cdots & C_1 H_M^T H_M + 2\alpha_M \end{bmatrix}_{ML \times ML}, \quad (\text{A4})$$

$$T_h = [H_1^T Y_{\text{output}} \ H_2^T Y_{\text{output}} \ \cdots \ H_M^T Y_{\text{output}}]_{ML \times ML}. \quad (\text{A5})$$

式中的 C_1 和 C_2 可由式(A6)和(A7)求得:

$$C_1 = 1 - 2\lambda(1 - \frac{1}{M})^2, \quad (\text{A6})$$

$$C_2 = 2\frac{\lambda}{M}(1 - \frac{1}{M}). \quad (\text{A7})$$

惩罚系数 λ 、集成规模 M 以及正则化参数 α 均由实验给定.

H 为隐藏层输出矩阵, 即

$$H = \begin{bmatrix} G(w_1 \cdot X_{\text{input}}^1 + b_1) & \cdots & G(w_L \cdot X_{\text{input}}^1 + b_L) \\ \vdots & \ddots & \vdots \\ G(w_1 \cdot X_{\text{input}}^N + b_1) & \cdots & G(w_L \cdot X_{\text{input}}^N + b_L) \end{bmatrix}_{N \times L}. \quad (\text{A8})$$

通过上述式子可求出全局输出权值:

$$\beta_{ens} = H_{corr}^{-1} T_h. \quad (A9)$$

A2) 各产品收率模型参数设置.

本文分别采用120组训练样本, 训练集成随机权神经网络, 获取汽油、柴油、液化气的收率模型. 此外, 分别采用47组测试样本来验证各收率模型的有效性. 不同产品收率模型的输入均为催化裂化反应再生过程操作优化时的决策变量值, 即原料油预热温度, 原料油流量, 反应温度, 再生温度, 再生压力, 反应压力, 提升蒸汽流量等7个操作变量值. 各收率模型中的隐藏层节点个数 L , 集成规模 M , 惩罚系数 λ 和正则化参数 α 均采用枚举法确定, 其分别取7, 8, 0.6, 0.22.

A3) 各产品收率模型的仿真结果.

汽油、柴油和液化气收率模型的47组测试样本的相对误差曲线分别如图A2-A4所示. 由图A2可知, 汽油收率测试样本的相对误差最大为2.5%, 且绝大多数样本的相对误差集中在0.2~1.8%; 由图A3可知, 柴油收率测试样本的相对误差最大为1.38%, 其绝大多数样本的相对误差集中在0~1%; 由图A4可知, 液化气收率测试样本的相对误差最大为3.8%, 其绝大多数样本的相对误差集中在0~2.4%; 这说明所建立的汽油、柴油和液化气收率模型均具备较好的精度和泛化能力.

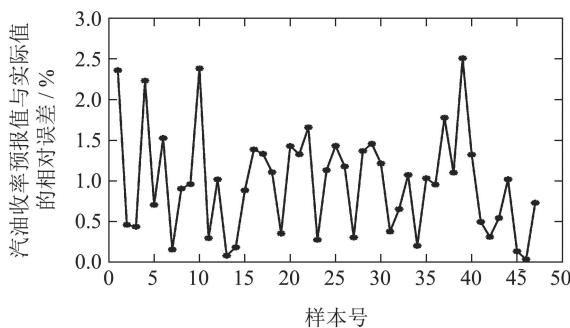


图 A2 集成随机权神经网络结构图

Fig. A2 The structure chart of ensemble neural networks with random weights

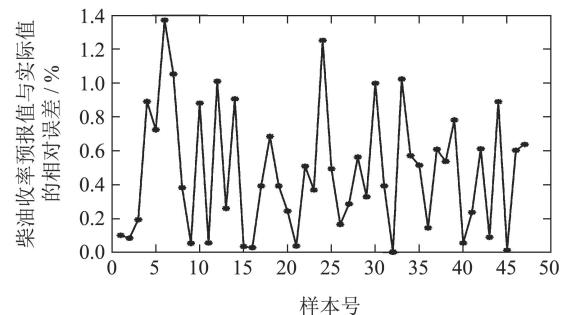


图 A3 集成随机权神经网络结构图

Fig. A3 The structure chart of ensemble neural networks with random weights

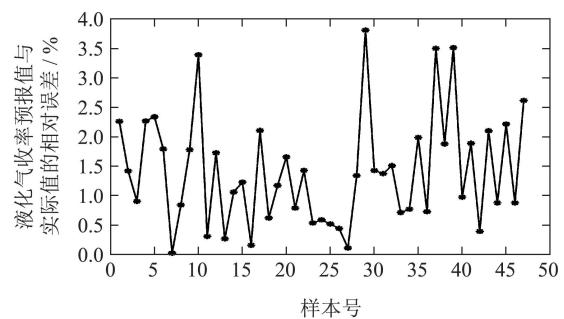


图 A4 集成随机权神经网络结构图

Fig. A4 The structure chart of ensemble neural networks with random weights

作者简介:

曹 蕃 硕士研究生, 目前研究方向为催化裂化装置建模与操作优化, E-mail: 1206592069@qq.com;

陈庆达 博士研究生, 目前研究方向为进化计算在复杂工业过程中的应用, E-mail: cqd0309@126.com;

丁进良 博士生导师, 目前研究方向为复杂工业过程智能建模与智能优化与控制、生产全流程运行优化、计算智能及其应用研究, E-mail: jlding@mail.neu.edu.cn.