11 周文琼,李庆忠,范路桥等.SaaS 模式多租户数据存贮模型的研究与实现.计算机科学,2013,47(10):194–197.

改进的细菌觅食优化算法①

吕 振, 沈建国

(辽宁工程技术大学 电气与控制工程学院, 葫芦岛 125105)

摘 要: 细菌觅食优化算法是一种受大肠杆菌觅食现象启发产生的一种群体进化算法, 该算法具有良好的全局优化能力, 鲁棒性强, 算法简单等优点, 但其也存在易早熟, 收敛速度慢等缺点. 根据其缺点, 提出了一种改进的细菌觅食优化算法, 改进后的算法收敛速度加快, 在一定程度上避免了易早熟的缺点. 将原算法和改进算法应用于 PID 参数的在线自整定, 通过 matlab 仿真实验证明了算法改进后的优越性.

关键词: 细菌觅食优化算法; 趋药性行为; 聚集行为; 繁殖行为; 迁徙行为; ABFO; PID 控制

Bacterial Foraging Optimization Algorithm of Improvement

LV Zhen, SHEN Jian-Guo

(Institute of Electrical and Control Engineering, Liaoning Technical University, Huludao 125105, China)

Abstract: Bacteria foraging optimization algorithm is a kind of group evolution algorithm inspired from E. coli foraging phenomenon, and it has good global optimization, strong robustness, simple arithmetic, and so on. But it is also easy to premature and slow convergence speed and other shortcomings. According to its shortcomings, it is proposed that an improved bacteria foraging optimization algorithm. The improved algorithm accelerates the convergence speed and avoids the premature in a certain extent. It is applied to on-line self-tuning of PID parameters that the original algorithm and improved algorithm. And through the matlab simulation, it is proved that the superiority of the improved algorithm.

Key words: bacteria foraging optimization algorithm; chemo taxis; swarming; reproduction; elimination and dispersal; ABFO; PID control

1 引言

在自然界中, 有着群体生活的生物, 大都具有完成复杂行为的能力. 人们受到这些群体生物的启发, 总结出一些仿生的智能优化算法. 仿生的智能算法中的每一个生命体都具有智慧和经验, 生物体相互之间通过某种作用机制解决复杂的问题. 通过模拟人类大肠杆菌觅食行为而来, 美国俄亥俄州立大学 Kevin M.Passino 教授于 2002 年提出了细菌觅食优化(BFO)算法[1]. 该算法具有良好的全局优化能力, 鲁棒性强, 算法简单等优点, 但其收敛速度慢且易早熟. 因此, 需要对算法进行改进, 文献[2]结合“和声搜索”算法对溢出空间的菌体进行处理使算法收敛速度加快, 文献[3]通过引入驱散过程加强了算法全局寻优能力, 文献[4]用基因算法和微分进化算法改进BFO算法, 提高了搜索全局最优的能力. 在此, 通过对细菌觅食优化算法中各种行为的详细分析, 引入细菌间作用力范围, 提出了改进的细菌觅食优化算法.

2 细菌觅食优化算法

在 BFO 算法中包含趋药性行为、聚集行为、繁殖行为和迁徙行为等四大步骤. 其详细介绍如下:

2.1 趋药性行为

在大肠杆菌觅食过程中, 其自身的控制系统会向着营养丰富的区域, 这被称为细菌的趋药性行为, 游走和旋转是其基本动作. 游走是按照前一步方向进行运动, 而旋转是换个方向运动. 细菌的趋药性行为就

① 基金项目:国家自然科学基金(51274118)
收稿时间:2013-10-28;收到修改稿时间:2013-11-12

是模拟这两种基本动作. 其行动描述为: 先在随机方向上运动一步, 若此位置上的适应值高于原位置, 那么下一步就在这个方向上继续前进, 反之, 则进行旋转运动. 每个细菌的趋药性运动都有最大尝试数.

2.2 聚集行为

聚集行为是菌群觅食的过程中, 通过群体间的相互作用来实现群体的聚集. 用向量 $X_i=(x_1,x_2,\cdots,x_v)$ 来表示第 i 个细菌的位置信息, $J(X_i)$ 来表示细菌在这个位置上的适应度函数, 那么聚集作用在整个细菌群体所处位置的数学表达式为:

$$J_{cc}(X_i)=\sum_{i=1}^{N}J_{cc}^{i}=\sum_{i=1}^{N}J_{attract}^{i}+\sum_{i=1}^{N}J_{repellant}^{i} \quad (1)$$

其中 J_{cc} 表示引力和斥力的合力效果, $\sum_{i=1}^{N}J_{\text{attract}}^{i}$ 与 $\sum_{i=1}^{N}J_{\text{repellant}}^{i}$ 分别表示引力部分和斥力部分的作用.

2.3 繁殖行为

在经过趋药性行为后, 按照细菌的健康度来淘汰觅食能力弱的细菌, 但为了维护种群的规模, 剩下的细菌将会繁殖. 一般把吸收到多少营养来判断细菌的健康度, 在这里, 健康度被定义为细菌在趋药性行为中所经历的各位置适应值的代数和.

对整个菌群按照健康度进行排序, 细菌健康度较低的 N/2 个个体被淘汰, 剩下的 N/2 个细菌进行繁殖行为. 一个产生两个相同的子代, 而子代继承了父代的位置、步长以及其它信息. N 为菌群数量, 一般取为偶数.

2.4 迁徙行为

迁徙行为是实际生活中的细菌外界环境发生变化时自身死亡或者跳转到新区域的现象, 是强迫菌群离开原来位置的方法. 在算法中, 菌群繁殖几代后, 细菌以概率 P_{ed} 被重新分布到寻优区间. 这里 P_{ed} 被称为迁徙概率.

整个算法的流程如图 1 所示.

通过分析 BFO 算法可知:

(1) 由公式(1)可知通过修正细菌适应度函数, 使细菌被动地实现聚集行为. 此方法简单易行, 但会改变适应度函数的分布.

(2) 如果细菌本身处于适应度较差的位置, 此时得到一个引力的修正信号, 那么它将会取得更优的适应值, 但这样会引入干扰.

(3) 在 BFO 算法寻优的最后阶段, 菌群都聚集到了全局最优值附近, 这时细菌间的斥力成为一个不利因素, 它会阻碍菌群聚集. 若细菌相互之间只存在引力, 则收敛速度大大加快, 可能导致早熟收敛.

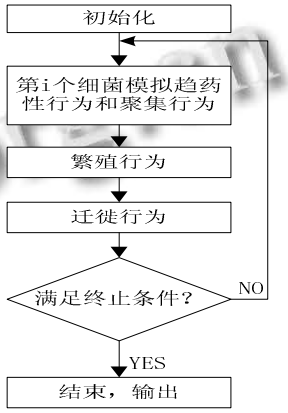

图 1 BFO 算法流程图

针对原菌群算法存在的问题, 提出了改进的细菌觅食优化(Advanced Bacterial Foraging optimization, ABFO)算法.

3 ABFO算法分析

通过对蚁群群算法、遗传算法等多种仿生算法的学习[5-8], 提出了 ABFO 算法. 为了阐述问题方便, 下面对 ABFO 算法中用到的符号含义解释如下:

N 表示菌群的个体数量, v 表示菌群算法优化的维度, $X_i=(x_1,x_2,\cdots,x_v)$ 表示第 i 细菌所处的位置, 其中 $x_i(i=1,2,\cdots,v)$ 表示细菌在第 i 维上的坐标; $x_i\in[x_{i\min},x_{i\max}],i=1,2,\cdots,v$, 为各个分量上细菌坐标的取值范围; J=f(X)表示细菌当前所在位置 X 处的营养浓度, J 为适应度函数; step(i,j)表示第 i 个细菌在第 j 个维度上的步长. limit 表示细菌的作用力范围, $\lim it_i$ 表示细菌在第 i 维度上的作用力范围大小. limitattract 表示菌体间引力的作用范围, limitrepell 表示菌体间斥力的作用范围. trynum 是细菌在一次趋药性行为中的最大尝试数.

3.1 细菌的作用力范围

细菌在每一个维度上的作用力范围是不同的, 通

常是将所有维度中取值范围最小值作为细菌的作用力范围. 为提高搜索的速度和效率, 对细菌的作用力范围定义如下:

设细菌处于一个 V 维空间中, 其位置信息由向量 $X(x_1,x_2,\cdots,x_v)$ 示, 如果对于空间中的另一随机位置 $X(x_1,x_2,\cdots,x_v)$, 有:

$$|y_i-x_i|\le \lim it_i, i=1,2,\cdots,v \quad (2)$$

那么, 就称Y在X的作用力范围之内, 其中 $\lim it_i$ 为第 i 个分量上的作用力范围, 代表引力范围或者斥力范围.

将作用力分为引力与斥力之后, 引力范围和斥力范围向量来表示. 步长的取值问题与维度问题同理. 因此为达到满意的搜索效果, 对各个维度采用各自不同的步长来进行搜索. 令 step(i,j)是第 i 个细菌在第 j 个维度上的步长, 那么对于一个细菌群体, 种群规模为 N, 维度为 v, 整个细菌群体的步长就可由矩阵 $step=(step(i,j))_{N\times v}$ 所表示; 如果整个细菌群体都采用相同的步长, 则步长可用向量 $step=(step_1,step_2,\cdots,step_v)$ 来表示, 其中 $step_i(i=1,2,\cdots,v)$ 为对应维度上的步长.

令 $\lim it_j$ 为第 i 个细菌在第 j 个维度上的作用力范围, 那么步长 step(i,j)与的初始值应满足等式(5)如下:

$$step(i,j)\cdot trynum\le \lim it_j, j=1,2,\cdots,v \quad (3)$$

其中trynum用来表示为细菌在一次趋药性行为中最大尝试数, 式(3)可以保证细菌在每一次趋药性行为中, 都不会走出作用力得范围; 因为 $\lim it$ 引力范围比斥力范围要大, 因此即为引力的作用范围 $\lim itattract_j$.

若令第 i 个维度上细菌的引力范围为 $\lim itattract_i$, 细菌的斥力范围为 $\lim itrepell_j$, 则引力范围和斥力范围的初始值应该满足如下公式[9]:

$$\lim itrepell_i=\alpha\cdot\lim itattract_i \quad (4)$$

其中 α 的取值范围一般在[0.01, 0.2]之间. 公式(4)的是为了设定引力与斥力区域的比例, α 表示斥力范围相对引力范围比值.

3.2 趋药性行为的改进

在BFO算法趋药性行为中, 采取了先让细菌往某一随机方向运动一步, 再计算其适应值的策略. 如果在细菌在觅食过程中原位置是所遇到的最优值, 就会丢掉这个历史最优位置的信息. 另外趋药性行为还存在一个问题: 细菌在经历过一次趋药性行为后会进入到一个不可行域. 因此在一次趋药性行为的末尾有必要加入溢出检查环节, 让溢出的细菌返回到初始化时所限定的区域 $x_i\in[x_{i\min},x_{i\max}],i=1,2,\cdots,v$. 设 temp 为一个计数器, 用来记录尝试数目是否达到 trynum; count 为一个判定标识, 当 $count\neq 0$ 时表示找到了比初始位置更好的位置; X_j 表示细菌的试探性新位置, X_{next} 为在趋药性行为中细菌的最优位置. 则改进的趋药性行为流程如下:

(1) 假定第 i 个细菌当前位置为 $X_i(x_1,x_2\cdots,x_v)$, 计算细菌在此位置上的适应度函数值 $J(X_i)$, 令 $J_{last}=J(X_i)$ 并它记为细菌 i 的历史最优值;

(2) 然后会产生一个随机方向向量, 将其单位化, 记作 $\Delta(\Delta_1,\Delta_2,\cdots,\Delta_v)$; 计数器 temp 初值为 0, 如果 count=0, 则在 Δ 方向前进一步, 同时 temp++; 得到细菌的试探新位置 X_j 用公式(5)所示第 j 个分量上的计算公式得 X_j 的坐标:

$$x_j=x_i+\Delta_i\cdot step(i,j) \quad (5)$$

上式中 x_j 表示新位置 X_j 的第 j 个分量, 而 x_i 表示当前位置 X_i 的第 j 个分量.

(3) 计算 X_j 位置上的适应度函数值 $J_j=f(X_j)$. 如果有 J_j 优于 J_{last} 那么表明在第(2)步中确立的方向是正确的, 可以令细菌自身的适应值得以改善, 因此令 $J_{next}=J_{last}=J_i$, $X_{next}=X_j$, 判定标识 count++, 然后继续在此 Δ 方向上游走, 每游走一次计数器 temp++, 直到 J_i 不再优于 J_{last} 为止, 跳出循环; 如果 J_j 差于 J_{last}, 且计数器 temp<trynum, 那么跳转步骤(2).

(4) 当计数器 temp 达到最大尝试次数 trynum 时, 依然没能找到比 J_{last} 更好的值, 则往随机方向迈一步.

(5) 越界检查, 输出 J_{next}, X_{next}.

3.3 聚集行为的改进

通过对细菌间相互作用的机制的相应调整, 并应用在聚集行为. 则改变后的聚集行为的程序流程如下:

(1) 求得细菌 i 在位置 $X(x_1,x_2\cdots,x_v)$ 的适应值 $J(X)$;

(2) 求得满足(6)式所示引力范围之内的所有的点 Y;

$$Y(y_1,y_2,\cdots,y_v)=Y(y_1,y_2,\cdots,y_v)\,|\,|y_i-x_i|\le \lim itattract_i,\ i=1,2,\cdots,N \quad (6)$$

其中 $\lim itattract_i$ 为细菌在第 i 个维度上的引力作用范围, N 为细菌的种群数量.

(3) 求出满足(6)式并且位置为 X 的细菌数 n, 然

后求得菌群中适应度最高的细菌的坐标, 再讨论目标函数的最大值, 所需的最优值就是函数的最大值, 如(7)式:

$$Y_{\max}(y_1, y_2, \cdots, y_v) = Y \,||\, \max(J(Y(y_1, y_2, \cdots, y_v))) \quad (7)$$

(4) 计算细菌在 $Y_{\max}$ 处的适应值 $J(Y_{\max})$. 如果 $J(Y_{\max})$ 大于 J(X), 那么执行下面公式:

$$X_{next} = \overline{XY_{\max}} \cdot C \cdot rand \cdot rolla \cdot rollr \quad (8)$$

其中 $\overline{XY_{\max}}$ 表示从 X 指向 $Y_{\max}$ 的方向向量: C 为加速因子, 其初值一般令 C=2; rand 为一个[0, 1]之间的随机数. Rolla 和 rollr 分别为引力和斥力的系数.

3.4 繁殖行为的改进

对于繁殖行为, 首先就是根据健康度对细菌进行排序. 把细菌在趋药性和聚集两种行为中适应度的代数和作为此细菌的健康度. 为计算方便把单个细菌在所经历的寻优过程中的最优的适应度作为菌体的健康度. 因为趋药性行为和聚集行为就能保证第 i 个细菌在当前所处的位置 X_i 处的适应值 $J(X_i)$ 是其自身历史最优值, 因此健康度函数可定义为:

$$Health(i) = \max(J_i) = J(X_i) \quad (9)$$

繁殖的行为有很多方式, 进行繁殖的细菌数目也可自行选择. 为了计算方便, 把种群数量 N 取为偶数, 淘汰 N/2 个健康度较低的细菌, 剩下的进行繁殖, 从而不改变种群规模.

在改进的繁殖行为中, 为实现更精密地搜索优化空间, 子代遗传父代的步长将要变小, 如公式(10)所示.

$$step(k+1) = step(k) \cdot \beta \quad (10)$$

$\beta \in [0,1]$表示子代步长为父代步长的$100\beta\%$.

3.5 迁徙行为的改进

对 BFO 算法的迁徙行为做了如下调整: 一是调整步长: 在繁殖行为中, 子代的步长比父代变得更小了, 会减慢子代搜索的速率. 为此, 可令子代的步长存在一定几率重置为某父代的步长, 称为“返祖”现象. 二是调整新加入参数, 如加速因子 C, 为使改进的算法有更好的性能, 调整方法可在在算法的后半段令 C=1, 减小算法的“粗调”因素.

改进的细菌觅食算法比原算法在很多方面做了进一步的改善, ABFO 算法的整个流程图如图 2:

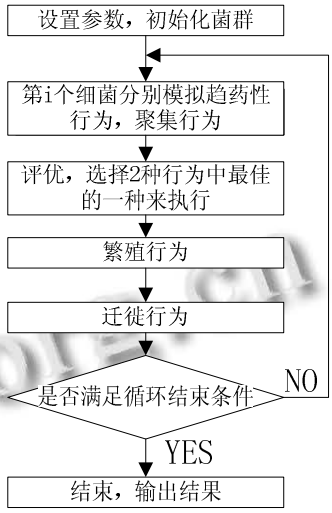

图 2 ABFO 算法流程图

4 细菌优化算法在PID控制中的仿真实验

4.1 ABFO 在 PID 中的应用

PID 控制器在工业生产中应用相当广泛的控制技术, 现在将 ABFO 算法和 PID 控制器结合在一起, 选取典型的被控对象, 实现对输入信号的在线跟踪[10]. 加入 ABFO 算法的 PID 控制框图如图 3.

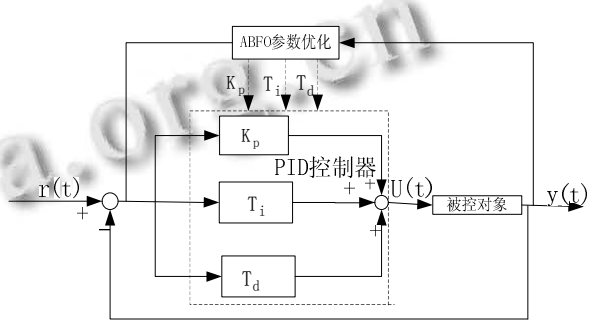

图 3 ABFO-PID 控制框图

r(t)为系统输入; y(t)为系统输出, U(t)为控制器输出.

PID 控制系统调节器 C(s)传递函数为:

$$C(s) = K_p(1 + \frac{1}{T_i s} + T_d s) \quad (11)$$

所选取的的控制对象的传递函数如下:

$$G(s)=\frac{e^{-s}}{(s+1)^2} \tag{12}$$

对于此控制系统，采用 ITAE 性能指标[11]来作为算法的适应度函数，来控制回路的 PID 参数进行优化设计，ABFO 算法流程如下:

(1) 设细菌的群体规模 N=20，引力的 范围向量 limitattract=(0.3, 0.2, 0.05)， 斥力的范围向量 为 0 向量；K_p，T_i，T_d 为菌群算法 所优化的三个 参数，那么细菌个体的位置为 3 维向量$[K_p,T_i,T_d]$，各个分量 的 取 值 范 围 [12] 为： $K_p\in[1.4034,4.2176]$, $T_i\in[0.825,2.433]$, $T_d\in[0.2039,0.6123]$；从上述可行域内随机选取 N 个细菌，组成原始菌群. step(i,j)为单个细菌步长，其中在 3 个维度上细菌步长的初始化区域分别为 $step(i,1)\in[0.02,0.2]$,$step(i,2)\in[0.01,0.8]$, $step(i,3)\in[0.02,0.2]$,然后在上述区域内随机生成 N 个细菌所对应的上述 3 个维度上的步长，形成原始的步长矩阵 $step=(step(i,j))_{N\times3}$；其它参数的取值为: trynum=6, C=2.

(2) 选择趋药性行为和聚集行为适应度最优来实际执行. 设置一个迭代次数计数器 Gen=0，执行一次评优，则视为迭代一次，算法的最大迭代次数为 $Gen_{\max}=400$.

(3) 设每经过 100 次迭代后，菌群进行繁殖行为；采用冒泡排序方法对菌群的健康度进行排序，对健康度高的一半数量的细菌进行繁殖，并令子代的步长为父代的 0.5 倍.

(4) 繁殖行为后，则执行迁徙行为，菌群以迁徙概率 P_{ed} =0.1 重新分配到另一个寻优空间中去.

(5) 终止条件判断: 判断最大迭代次数 $Gen_{\max}$ 是否达到预置值 400，若是，则将计算结果输出；否则跳转步骤(2).

4.2 ABFO 算法仿真

用 Visual C++6.0 软件进行编程，MATLAB 软件进行仿真. 在仿真实验中，系统的输入采用单位阶跃函数，取 T=0.1 秒为采样周期，采样次数取为 500 次. 控制系统的系统结构图如图 3 所示，采用优化算法对控制系统的控制对象进行优化. 对控制系统中的 PID 参数，使用 ABFO 算法进行优化，并将优化结果同 BFO 算法优化结果在性能上作比较，如表 1 所示.

表 1　BFO 和 ABFO 控制性能比较

PID 设计方法	K_p	T_i	T_d	$\sigma\%$	$t_s(s)$	$t_p(s)$	$t_r(s)$	ITAE	优化时间（秒）
BFO	1.8998	1.8205	0.5958	3.3007	4.5	2.9	0.83	1.459	42.551
ABFO	1.4537	1.9893	0.5679	0.2057	5.2	9.1	1.97	1.348	18.93

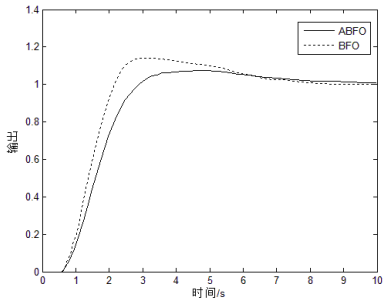

图 4　两种算法的单位阶跃响应曲线

表 1 中，t_p 为峰值时间、t_r 为上升时间、t_s 为调节时间. 从中能够看出，ABFO 算法与 BFO 算法相比，其具有更快的运行速度，性能评价指标 ITAE 的取值更小，超调量变小. 从中可以看出 ABFO 的优势. 将上述两种算法所优化的 PID 参数带入对应的系统中，分别对两个系统画阶跃响应曲线，如图 4 所示.

从图 4 中可以更直观的看到经 ABFO 算法所优化的系统相比于 BFO 算法所优化的系统具有更小的超调量，更快的反应速度，对控制参数的整定效果更好，ABFO 算法的优越性得到验证.

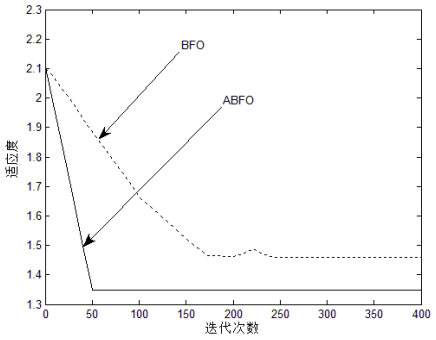

图 5　两算法迭代图

同时作出BFO和ABFO算法中适应度函数随迭代次数的变化曲线如图 5. 从图中可以看出 ABFO 算法在收敛的速度和运算的精度上均优于 BFO 算法, 并且避免了早熟, 故可以得出改进算法的合理, 有效.

5 结论

通过对 BFO 算法的四种行为的详细介绍和分析, 发现了其存在收敛速度慢、稳定性差的问题, 为此提出了改进的细菌觅食优化算法, 对算法中的四种行为做了些改变. 在趋药性行为中, 增加了模拟试探, 可以保留住历史最有位置信息, 在趋药性行为的末尾加入了溢出检测环节, 让溢出细菌回到初始化所在区域. 对于聚集行为, 引入了加速因子, 提高了收敛速度. 在繁殖行为中, 子代以较小的步长遗传父代, 并将历史的最优适应度作为健康以方便计算. 在迁徙行为中, 如果父代最开始的步长是合适的, 那么子代细菌就可以通过返祖现象将步长重置回去. 最后将 BFO 和 ABFO 这两种算法应用在 PID 控制中, 通过 matlab 仿真实验进行对比, 结果得出 ABFO 算法具有更高的精度和更快的收敛速度, 优于 BFO 算法.

参考文献

1 Passino KM. Biomimicry of bacterial foraging for distributed optimization and control. Control Systems, IEEE, 2002, 22(3):52–67.

2 张璨,刘锋,李丽娟.改进的细菌觅食优化算法及其在框架结构设计中的应用.工程设计学报, 2012,19(6):422–427.

3 马溪原,吴耀文,方华亮,等.采用改进细菌觅食算法的风/光/储混合微电网电源优化配置.中国电机工程学报,2011, 31(25): 17–25.

4 李珺,党建武,卜锋.细菌觅食优化算法的研究与改进.计算机仿真,2013(4):344–347.

5 Das S, Dasgupta S, Biswas A, et al. On stability of the chemotactic dynamics in bacterial-foraging optimization algorithm. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Trans. on, 2009, 39(3): 670–679.

6 催志华,曾建潮.微粒群优化算法.北京:科学出版社,2011: 70–72.

7 周雅兰.细菌觅食优化算法的研究与应用.计算机工程与应用, 2010,46(20):16–21.

8 李明,杨成梧.细菌菌落优化算法.控制理论与应用,2011, 28(2):223–228.

9 胡洁.细菌觅食优化算法的改进及应用研究.武汉:武汉理工大学,2012.

10 金建平.一种新型的过程模型参数辨识方法.自动化技术与应用,2012,31(1):4–7.

11 樊非之.菌群算法的研究及改进.北京:华北电力大学, 2013.

12 柯晶,李威武,钱积新.基于粒子群优化的时变系统辨识.系统工程与电子技术,2003,25(10):1256–1259.