

DOI: 10.3969/j.issn.1006-4729.2016.02.013

基于改进细菌觅食算法的无功优化

邓集瀚, 杨俊杰

(上海电力学院 电子与信息工程学院, 上海 200090)

摘要:采用改进的细菌觅食(MBFO)算法求解电力系统无功优化问题,引入了步长递减的控制策略,改善了算法前期的全局搜索能力和后期的局部搜索能力;引入了SA-PSO变异算子,从而使个体可以相互交流,并从精英那里得到经验;引入遗传算法的交叉和赌盘选择,保护了精英个体,同时降低了解劣化的概率.以IEEE-30节点为例的算例结果表明,较其他几种优化方法而言,MBFO具有更快的收敛速度和更好的优化效果,故该算法在解决无功优化问题上可行且有效.

关键词:电力系统;无功优化;细菌觅食算法

中图分类号:TM714.3;TP18

文献标志码:A

文章编号:1006-4729(2016)02-0167-08

Reactive Power Optimization Based on Modified Bacterial Foraging Optimization Algorithm

DENG Jihan, YANG Junjie

(School of Electronic and Information Engineering, Shanghai University of Electric Power, Shanghai 200090, China)

Abstract: The reactive power optimization problem of power system is solved by Modified Bacterial Foraging Optimization Algorithm. The algorithm introduces a decreasing step sizes control mode to improve the global search ability in the early stage and the local search ability in the late stage. SA-PSO mutation operator is introduced to make individuals learn from each other; Genetic Algorithm roulette choosing and crossover mutation is introduced to protect the elites and suppress the degeneracy phenomenon. Taking IEEE-30 node test system for examples, the results show that the algorithm is effective in solving reactive power optimization problems and prove that the algorithm convergence speed and optimization algorithm have better performance than other optimization results.

Key words: power system; reactive power optimization; bacterial foraging optimization algorithm

电力系统无功优化指的是在满足所有约束条件的前提下,当系统拓扑结构和运行状况给定时,通过调节某些控制变量,从而改变系统的潮流分

布,并使得某些性能指标最优.无功优化是最优潮流的一个分支,并且是一个典型的多变量、非线性、多约束的组合优化问题.现阶段求解无功优化问题

收稿日期:2015-07-07

通讯作者简介:邓集瀚(1989-),男,在读硕士,湖南邵阳人.主要研究方向为电力系统优化运行. E-mail: napoleon6588@sohu.com.

的方法主要有经典方法和智能优化算法两种。

经典方法的典型代表有梯度算法 (Gradient Algorithm)、牛顿法 (NEWTON Method)、内点法 (Interior Point Method), 该类方法的核心在于采用梯度信息进行寻优。智能优化算法具有较好的鲁棒性, 并且对研究对象是否可微或可导不敏感, 求解无功优化的智能优化算法有粒子群算法^[1]、鱼群算法^[2]、蚁群算法^[3]、免疫算法^[4]、差分进化算法^[5]、细菌觅食算法^[6-10]等。细菌觅食 (Bacterial Foraging Optimization, BFO) 算法是由 PASSION K M^[11]提出的一种仿生算法, 该算法跟其他智能优化算法同样存在容易陷入局部最优而早熟的不足。为提高细菌觅食算法的全局搜索能力, 本文对该算法的趋化步长、群体感应机制、驱散操作进行了改进, 形成了一种改进的细菌觅食 (Modified Bacterial Foraging Optimization, MBFO) 算法, 并通过 IEEE-30 节点系统的算例分析验证了该算法的可行性和有效性。

1 无功优化的数学模型

无功优化数学模型由目标函数、等式约束条件、不等式约束条件 3 部分组成。

1.1 目标函数

本文以网损最小为目标, 目标函数为:

$$\min P_{\text{loss}} = \sum_{i,j} (U_i^2 + U_j^2 - 2U_i U_j \cos\theta_{ij}) g_{ij} \quad (1)$$

式中: U_i, U_j ——节点 i 和节点 j 的电压幅值;
 $\cos\theta_{ij}$ ——节点 i 和节点 j 间的相角差;
 P_{loss} ——网损;
 g_{ij} ——支路电导。

将 PQ 节点电压和发电机无功功率通过罚函数引入目标函数, 从而式(1)可改写为:

$$\left\{ \begin{aligned} \min F &= \min P_{\text{loss}} + \lambda_1 \sum_{i=1}^{N_{\text{pq}}} \left(\frac{U_i - U_{i\text{lim}}}{U_{i\text{max}} - U_{i\text{min}}} \right)^2 + \\ &\quad \lambda_2 \sum_{i=1}^{N_{\text{pu}}} \left(\frac{Q_i - Q_{i\text{lim}}}{Q_{i\text{max}} - Q_{i\text{min}}} \right)^2 \\ U_{i\text{lim}} &= \begin{cases} U_{i\text{max}}, U_i > U_{i\text{max}} \\ U_i, U_{i\text{min}} \leq U_i \leq U_{i\text{max}} \\ U_{i\text{min}}, U_i < U_{i\text{min}} \end{cases} \\ Q_{i\text{lim}} &= \begin{cases} Q_{i\text{max}}, Q_i > Q_{i\text{max}} \\ Q_i, Q_{i\text{min}} \leq Q_i \leq Q_{i\text{max}} \\ Q_{i\text{min}}, Q_i < Q_{i\text{min}} \end{cases} \end{aligned} \right. \quad (2)$$

式中: $U_{i\text{lim}}, Q_{i\text{lim}}$ ——当电压和发电机无功出力超过上下限时的设定值;

$U_{i\text{max}}, U_{i\text{min}}$ ——电压幅值的上限和下限;

$Q_{i\text{max}}, Q_{i\text{min}}$ ——发电机无功出力的上限和下限;

$N_{\text{pq}}, N_{\text{pu}}$ ——PQ 节点数目和 PU 节点数目;

λ_1, λ_2 ——PQ 节点电压越界和 PU 节点无功出力越界的惩罚因子。

1.2 等式约束

等式约束条件, 即常规潮流方程:

$$\begin{cases} P_i - U_i \sum_{j=1}^N U_j (G_{ij} \cos\theta_{ij} + B_{ij} \sin\theta_{ij}) = 0 \\ Q_i - U_i \sum_{j=1}^N U_j (G_{ij} \sin\theta_{ij} - B_{ij} \cos\theta_{ij}) = 0 \end{cases} \quad (3)$$

式中: P_i, Q_i ——节点 i 的注入有功功率和无功功率;

G_{ij}, B_{ij} ——线路的电导和电纳;

N ——总的节点数。

1.3 不等式约束

不等式约束由状态变量约束和控制变量约束构成。

控制变量约束为:

$$\begin{cases} U_{G_i, \text{min}} \leq U_{G_i} \leq U_{G_i, \text{max}}, i \in N_G \\ K_{T_i, \text{min}} \leq K_{T_i} \leq K_{T_i, \text{max}}, i \in N_T \\ Q_{C_i, \text{min}} \leq Q_{C_i} \leq Q_{C_i, \text{max}}, i \in N_C \end{cases} \quad (4)$$

式中: $U_{G_i}, K_{T_i}, Q_{C_i}$ ——发电机机端电压, 变压器变比, 补偿电容器容量;

$U_{G_i, \text{max}}, U_{G_i, \text{min}}$ ——发电机机端电压的上限和下限;

$K_{T_i, \text{max}}, K_{T_i, \text{min}}$ ——变压器变比的上限和下限;

$Q_{C_i, \text{max}}, Q_{C_i, \text{min}}$ ——补偿电容器容量的上限和下限;

N_G ——发电机数目;

N_T ——变压器的数目;

N_C ——补偿电容器数目。

状态变量约束为:

$$\begin{cases} U_{i\text{min}} \leq U_i \leq U_{i\text{max}} & i \in N \\ Q_{G_i, \text{min}} \leq Q_{G_i} \leq Q_{G_i, \text{max}} & i \in N_G \end{cases} \quad (5)$$

式中: Q_{G_i} ——发电机 i 的无功输出;

$Q_{G_i, \max}, Q_{G_i, \min}$ ——发电机 i 的无功输出的上限和下限.

2 细菌觅食算法的基本原理

根据文献[11]可知, BFO 主要由驱化、聚群、复制、驱散(迁徙)4个部分构成.

(1) 驱化(Chemotaxis) 大肠杆菌在寻找食物的过程中有旋转(tumble)和游动(swim)两种行为. 其操作方式如下: 细菌朝某随机方向游动一步; 若该方向的适应值低于未游动前位置对应的适应值, 则旋转, 并朝着另一任意方向游动, 若适应值增加, 则沿该方向继续前进; 若尝试次数到达上限, 则该细菌个体的趋向操作结束, 并选择下一个细菌进行趋向操作. 细菌 i 趋向操作的每一步如下:

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + \frac{C(i)\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \quad (6)$$

式中: $\theta^i(j, k, l)$ ——细菌 i 经过 j 次驱化, k 次复制, l 次驱散后的位置;

$C(i)$ ——细菌的趋化步长;

$\Delta(i)$ ——随机向量, 其分量均是 $[-1, 1]$ 上的随机数.

(2) 聚群(Swarming) 在吸引和排斥两种作用下, 细菌产生了聚群行为:

$$J_{cc}(\theta, P(j, k, l)) = \sum_{i=1}^S J_{cc}^i(\theta, \theta^i(j, k, l)) = \sum_{i=1}^S [-d_{\text{attract}} \exp(-w_{\text{attract}} \sum_{m=1}^p (\theta_m - \theta_m^i)^2)] + \sum_{i=1}^S [h_{\text{repellant}} \exp(-w_{\text{repellant}} \sum_{m=1}^p (\theta_m - \theta_m^i)^2)] \quad (7)$$

式中: $J_{cc}(\theta, P(j, k, l))$ ——群体感应机制下对单个细菌位置的修正值;

θ_m ——第 m 维中细菌的位置;

θ_m^i ——其他细菌的位置;

$d_{\text{attract}}, w_{\text{attract}}$ ——引诱剂的扩散距离和范围;

$h_{\text{repellant}}, w_{\text{repellant}}$ ——排斥剂的作用距离和范围.

(3) 复制(Reproduction) 驱化结束后, 进行复制操作. 健康度函数可表示为:

$$J_{\text{health}}^i = \sum_{j=1}^{N_c+1} J(i, j, k, l) \quad (8)$$

式中: $J(i, j, k, l)$ ——细菌 i 经过 j 次驱化, k 次复制, l 次驱散后的能量值.

用式(8)来计算细菌 i 在整个驱化过程中积累的能量, 健康度函数 J_{health}^i 越大代表细菌越虚弱, 淘汰掉 J_{health}^i 较大的那一半, 则其他细菌分裂, 新产生个体的位置与母体相同.

(4) 驱散(迁徙)(Elimination and Dispersal)

每个细菌产生一个随机数 P_{rand} , 若 $P_{\text{rand}} <$ 驱散概率 P_{ed} , 则该细菌被选中, 并在解空间重新初始化.

3 改进的细菌觅食算法及性能测试

3.1 MBFO 算法

3.1.1 改进的趋化步长

在使用 BFO 求解连续优化问题时, 通常设置为固定步长, 但是较小的步长会导致收敛缓慢, 甚至算法早熟; 较大的驱化步长则容易引起迭代后期强烈的震荡, 故对步长采用一种缩进的控制方式, 驱化步长随着迭代的进行而变得越来越小, 从而改善全局最优点周围的寻优能力.

$$C_{k+1}(i) = C_k(i)e^{-\tau} \quad (9)$$

式中: $C_k(i)$ ——细菌 i 的驱化步长;

τ ——衰减常量.

3.1.2 改进的群体感应机制

细菌之间通过一种特殊的化学物质来实现信息交换, 该物质称为自我诱导物. 细菌的生理行为会随着环境中自我诱导物的浓度而发生变化, 该现象就是群体感应效应.^[12-13]

聚群操作所计算的 J_{cc} 实质为群体感应机制作用下对个体位置的修正. 文献[14]至文献[16]指出群体感应机制对收敛速度和精度均能起到抑制作用.

粒子群(PSO)算法是一种颇为普及的智能仿生算法, 个体能够从近邻的发现中得到启发, 并从以往的经历中获得收益, 同时模拟褪火算法(SA)具有良好的局部搜索能力.^[17-18] 故将 PSO 和 SA 结合起来作为一个 PSO-SA 变异因子, 来改善细菌间的交流能力, 具体步骤如下.

步骤1 由 BFO 完成驱化操作后, 计算出个体最佳位置 P_{best} 和种群最佳位置 g_{best1} .

步骤2 利用当前的 g_{best1} 作为 SA 的初解, 并经过迭代后得到解 g_{best2} . 具体为: 根据初始解 g_{best1} 由 SA 状态函数形成新的解; 采用 Metropolis 准则计算接受该新解的概率; SA 是否抽样稳定, 不稳定则用状态函数重新生成解; 进行退温操作, 更新 g_{best2} .

步骤3 计算并比较 g_{best1} 和 g_{best2} 的适应度值的大小,并用较小数值对应的细菌位置作为当前的种群最佳位置.

步骤4 按照以下方程式,更新细菌的速度和位置

$$v_{id}^{new} = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} [v_{id}^k + c_1 r_1 (p_{bestid} - x_{id}^k) + c_2 r_2 (g_{besti} - x_{id}^k)] \quad (10)$$

$$\theta_{new}^i(j+1, k, l) = \theta_{old}^i(j+1, k, l) + v_{id}^{new} \quad (11)$$

式中: $\theta_{new}^i(j+1, k, l)$ ——第 i 个细菌经过 $j+1$ 次趋化、 k 次复制和 l 次迁徙后,并通过 PSO 更新后所对应的位置;

c_1, c_2 ——学习因子;

$\varphi = c_1 + c_2 > 4$;

g_{besti} —— g_{best1} 和 g_{best2} 的适应度中的最小值;

x_{id}^k, v_{id}^k ——细菌 i 在第 k 次迭代中第 d 维的位置和速度.

3.1.3 改进的驱散(迁徙)操作

在迁徙算子中,菌群里的每个细菌产生一个 $0 \sim 1$ 内的随机数 P_{rand} ,将它与恒定迁徙概率 P_{ed} 相比较,若 $P_{rand} < P_{ed}$,则对该细菌进行迁徙操作,即在寻优区间内进行初始化.文献[19]至文献[21]指出,尽管驱散机制能够提升算法跳出局部最优的能力,但是该机制对重新初始化后所产生新解的质量无任何判定,采用盲目的一律接收的方式,会造成部分已处于最优解附近的精英个体丢失,从而导致解的退化.

为了降低精英个体被选中的几率和防止重新初始化后产生新解的质量下降,引入遗传算法的赌盘选择思想和交叉机制.

赌盘选择可表示为:

$$P_{(i)} = \frac{|J_{health}^{min} - J_{health}^i|}{J_{health}^{max} - J_{health}^{min}} K P_{ed} \quad (12)$$

式中: J_{health}^i ——细菌个体的能量值;

$J_{health}^{max}, J_{health}^{min}$ ——种群中能量的最大值和最小值;

K ——修正系数.

当 $J_{health}^i = J_{health}^{min}$ 时, $P_{(i)} = 0$,从而使得 $P_{rand} < P_{ed}$ 不成立,实现了对精英个体的保护.

交叉机制即为:将选中的细菌所对应的位置记为 X_1 , X_1 重新初始化后产生的新位置为 X_2 , X_1 和 X_2 进行交叉操作得到 X_3 和 X_4 ,计算 X_i 对应的 J

值,并用最小 J 值对应的 X_i 替换掉 X_1 .

3.2 MBFO 算法流程

步骤1 初始化如下参数: $S, C(i), P_{ed}, \tau, K, N_s, N_c, N_{re}, N_{ed}, L, \alpha, T_0, c_1, c_2, r_1, r_2$. 其中, S 代表细菌的数目; $C(i)$ 为翻转步长; N_s 为最大游动长度; N_c, N_{re}, N_{ed} 分别代表驱化、复制、驱散操作的次数; L 为马尔可夫链长度; α 为降温系数; T_0 为初始温度; r_1, r_2 为 $(0, 1)$ 内的随机数.

步骤2 驱散循环: $l = l + 1$.

步骤3 复制循环: $k = k + 1$.

步骤4 驱化循环: $j = j + 1$. 菌群中的所有个体都按照细菌 i 进行如下的驱化操作.

(1) 按照 $X = x_{min} + P_{rand}(x_{max} - x_{min})$ 初始化细菌的位置,并计算初始适应度值 $J_{last} = J(i, j, k, l)$.

(2) 旋转,产生一个各分量均在 $[-1, 1]$ 内的向量 $\Delta(i)$.

(3) 移动细菌的位置,使得 $\theta(j+1, k, l) = \theta(j, k, l) + \frac{C(i)\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$,其中 $C(i)$ 按照式(9)变化.

(4) 计算适应度值 $J(i, j+1, k)$.

(5) 游动. 令 $m = 0$,若 $m < N_s$,则令 $m = m + 1$. 当 $J(i, j+1, k, l) < J_{last}$,更新 $J_{last} = J(i, j+1, k, l)$,使 $\theta(i, j+1, k) = \theta(i, j, k) + \frac{C(i)\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$,并计算新位置 $J(i, j+1, k, l)$; 否则令 $m = N_s$,同时游动结束.

(6) 选择下一个细菌进行驱化操作.

步骤5 引入 PSO-SA 变异因子,更新细菌的速度和位置.

步骤6 若 $j < N_c$,跳转步骤4.

步骤7 复制循环:计算细菌的健康度,将数值较高的淘汰掉,其余细菌则进行复制并产生与母体同位置的新个体.

步骤8 假如 $k < N_{re}$,则跳转步骤3.

步骤9 驱散循环:按本文提出的改进的驱散(迁徙)方式进行驱散.

步骤10 假如 $l < N_{ed}$,则跳转步骤2;否则算法结束.

3.3 MBFO 的性能测试

通过对4个标准函数的寻优来测试 MBFO 的性能,4个函数如表1所示.

表 1 测试函数

测试函数	函数表达式	寻优区间	最优解
Rosenbrock	$f_1(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	$[-15, 15]$	0
Rastrigrin	$f_2(x) = \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i) + 10)$	$[-5.12, 5.12]$	0
Griewank	$f_3(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]$	0
Ackley	$f_4(x) = 20 + c - 20e^{-0.2\sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}} - e^{\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)}$	$[-32, 32]$	0

本文共设计了 6 组算法对比测试实验,包括标准 BFO^[11]和几种改进的算法:双种群细菌觅食算法(DFBFO)^[22],量子细菌觅食算法(QBFO)^[23],差分进化-粒子群-细菌觅食混合算法(DE-BSO)^[24],差分进化细菌觅食算法(DE-BFO)^[25],算法参数参照相应文献.MBFO 参数

为: $S = 16, C = 0.02, P_{ed} = 0.25, \tau = 0.21, K = 1.36, N_s = 2, N_c = 25, N_{re} = 4, N_{ed} = 2, L = 10, \alpha = 0.91, T_0 = 1000, c_1 = c_2 = 2.1$,总迭代次数 = $N_c N_{re} N_{ed} = 200$ 代.各算法分别运行 20 次,取得的平均最优值如表 2 所示.MBFO 的收敛曲线如图 1 所示.

表 2 标准函数算法的优化效果比较

函数	维度	平均最优值					
		BFO	MBFO	DFBFO	QBFO	DE-BSO	DE-BFO
f_1	30	58.216	2.465e-9	27.337	1.815e-4	2.675e-1	1.845
f_2	30	17.038	2.916e-11	2.378e-2	7.832e-9	1.0e-5	1.65e-1
f_3	30	0.373	1.013e-7	6.606e-8	5.205e-11	1.0e-5	2.465e-4
f_4	30	4.324	2.330e-11	4.675e-6	5.382e-6	1.817e-4	9.533e-15

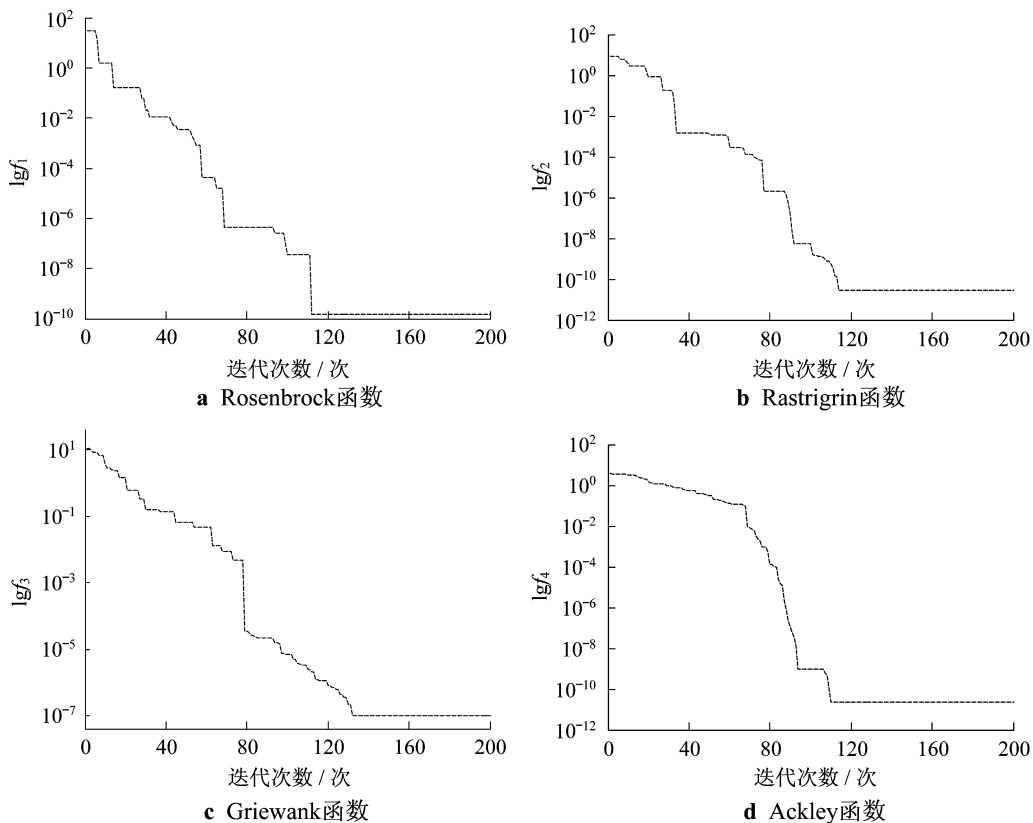


图 1 MBFO 的收敛曲线

由表2可以看出,对于函数 f_1 ,MBFO的解较BFO提升了9个数量级,对于函数 f_3 提升了7个数量级,对于函数 f_2 和 f_4 则提升了11个数量级,虽然在函数 f_3 和 f_4 上MBFO的优化效果比QBFO和DE-BFO差4个数量级,但是较标准BFO和其他大多数改进算法而言,MBFO仍获得较高的收敛精度。

4 基于MBFO的无功优化

运用MBFO算法求解电力系统无功优化问题的流程,如图2所示。

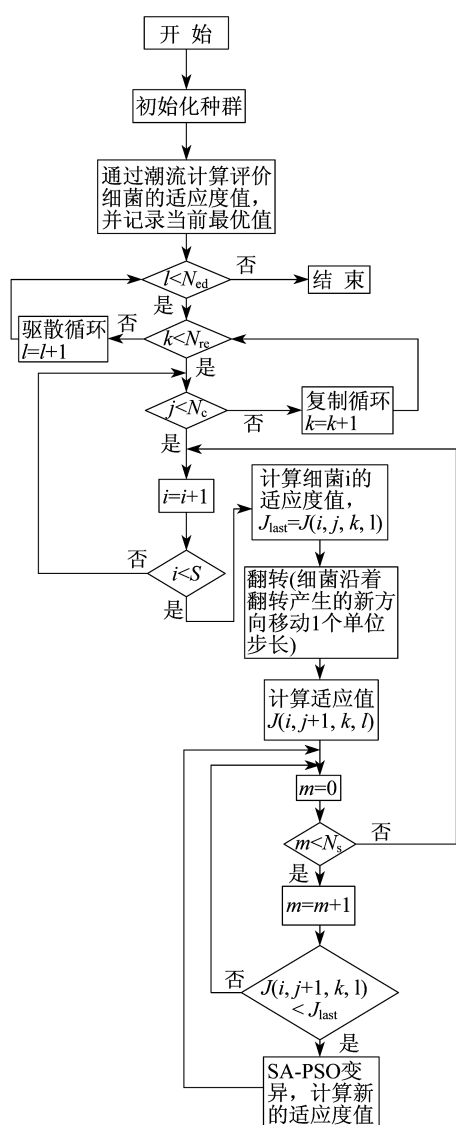


图2 基于MBFO算法的无功优化流程

5 算例分析

本文以IEEE-30节点系统为例,该系统有6

台发电机、21条负荷母线、41条支路、4台变压器、2个无功补偿节点,系统参数参见文献[26]。各状态变量上下限,各控制变量上下限及步长设置参见文献[27]。初始状态下,发电机的机端电压和变压器的变比均设置为1.0,经潮流计算系统总有功网损为 $P_{\text{loss}} = 0.05987$ 。

为了验证本算法的有效性,分别与标准遗传算法(SGA)^[28]和标准细菌觅食算法(BFO)^[8]进行比较。MBFO参数设置与上述相同,罚函数系数 $\lambda_1 = 6, \lambda_2 = 4$ 。各种算法分别运行30次,表3给出了各种算法所得到的最优结果,优化后各控制变量的值如表4所示。

表3 IEEE-30系统不同算法优化结果比较

算法	P_{loss} (p. u.)	P_{save} (p. u.)	网损下 降率/%	迭代次 数/次	计算时 间/s
SGA	0.05539	0.00448	7.483	71	62.07
BFO	0.05097	0.00890	14.866	56	45.91
MBFO	0.04836	0.01151	19.225	43	50.23

表4 控制变量优化后的值单位 p. u.

控制变量	节点号	SGA	BFO	MBFO
U_1	1	1.0750	1.0615	1.0782
U_2	2	1.0645	1.0658	1.0689
U_3	5	1.0423	1.0451	1.0466
U_4	8	1.0453	1.0459	1.0467
U_5	11	1.0337	1.0689	1.0729
U_6	13	1.0546	1.0633	1.0650
T_1	6~9	0.9400	0.9800	1.0400
T_2	6~10	1.0300	0.9900	0.9500
T_3	4~12	1.0400	1.0100	0.9700
T_4	28~27	1.0200	0.9900	0.9700
Q_1	10	0.3500	0.2700	0.1600
Q_2	24	0.0600	0.1600	0.1300

注: U_i —节点*i*的电压幅值, $i = 1, 2, 3, \dots, 6$; T_1, T_2, T_3, T_4 —变压器的档位; Q_1, Q_2 —电容的档位。

由表3可以看出,虽然采用MBFO进行无功优化所需时间比BFO多4.32s,但是得到的系统网损 P_{loss} 为0.04836,网损下降率为19.225%,显然这组结果优于其他几种方法,同时MBFO的迭代次数比其他两种算法少,故MBFO具有更快的收敛速度。综上可知,本文提出的方法有效、可行。

图3为SGA、BFO、MBFO3种算法进行无功

优化时取得的收敛曲线.由图3可知,SGA算法不但比BFO算法收敛慢,同时解的质量也不如后者;MBFO陷入局部最优时,寻优曲线呈现出一个平缓的直线区域,但是平缓区域的持续效果非常短暂,算法在较短的迭代次数内就能跳出局部最优陷阱并找到新的寻优发现,故MBFO具有较好的全局寻优能力,同时较其他两种算法而言,能够在更少的迭代次数内取得性能更好的解.

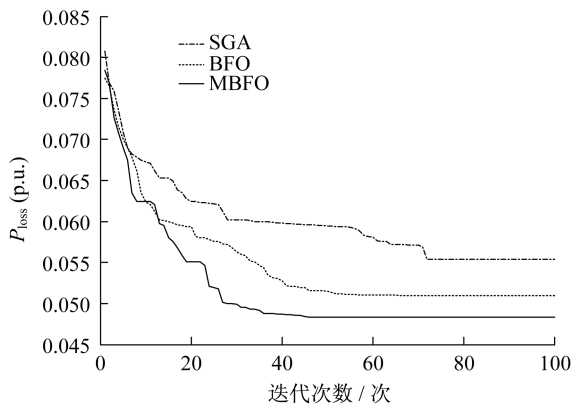


图3 3种算法的收敛曲线

6 结 语

对标准BFO算法中固定趋化步长、群体感应机制、驱散操作的不足进行分析,并逐条加以改进.6组测试函数寻优的对比试验结果说明,较其他5种算法而言,MBFO在大多数测试函数上具有更好的收敛精度.将MBFO算法用于电力系统无功优化问题,IEEE-30算例的仿真结果表明,较其他两种算法而言,MBFO具有更好的全局寻优能力和更快的收敛速度,因此本文提出的方法有效、可行.

参考文献:

- [1] 陈前宇,陈维荣,戴朝华,等.基于改进PSO算法的电力系统无功优化[J].电力系统及其自动化学报,2014,26(2):8-13.
- [2] 刘苏云,丁晓群.利用混沌鱼群算法的配网无功优化[J].电力系统及其自动化学报,2014,26(1):44-48.
- [3] 阮仁俊,陈焯,刘天琪.基于混沌理论和排序选择的蚁群无功优化算法[J].电网技术,2009,33(11):49-54.
- [4] 钟红梅,任震,张勇军,等.免疫算法及其在电力系统无功优化中的应用[J].电网技术,2004,28(3):16-19.
- [5] 曾学强,刘志刚,符伟杰,等.改进差分进化算法在电力系统无功优化中的应用[J].电网技术,2012,36(2):121-125.
- [6] 彭继慎,杨慕,紫马冰.含分布式发电的改进BFO算法配电网无功优化[J].计算机仿真,2015,32(5):127-131.
- [7] 李莹,简献忠,陈青.基于免疫进化细菌觅食算法的无功优化[J].上海理工大学学报,2014,36(3):245-249.
- [8] TRIPATHY M, MISHRA S. Bacteria foraging-based to optimize both real power loss and voltage stability limit[J]. IEEE Transactions on Power Systems, 2007, 22(1):240-248.
- [9] TANG W J, LI M S, WU Q H, *et al.* Bacterial foraging algorithm for optimal power flow in dynamic environments[J]. IEEE Transactions on Circuits and Systems, 2008, 55(8):2433-2442.
- [10] LI M S, TANG W J, TANG W H, *et al.* Bacterial foraging algorithm with varying population for optimal power flow[C]. Proc Evol Workshops, 2007:32-41.
- [11] PASSINO K M. Biomimicry of bacterial foraging for distributed optimization and control[J]. IEEE Control Systems Magazine, 2002, 22(3):52-67.
- [12] WILLIAMS P, CÁMARA M. Quorum sensing and environmental adaption in *Pseudomonas aeruginosa*: a tale of regulatory networks and multifunctional signal molecules[J]. Current Opinion in Microbiology, 2009, 12(2):182-191.
- [13] KUMAR L, CHHIBBER S, KUMAR R, *et al.* Zingerone silences quorum sensing and attenuates virulence of *Pseudomonas aeruginosa*[J]. Fitoterapia, 2015, 102(1):84-95.
- [14] TANG W J, WU Q H, SAUNDERS J R. Bacterial foraging algorithm for dynamic environment[C]. Proceedings of IEEE Conference on Evolutionary Computation, Canada, 2006:1324-1330.
- [15] TRIPATHY M, MISHRA S. Bacteria foraging based to optimize both real power loss and voltage stability limit[J]. IEEE Transactions on Power Systems, 2007, 22(1):240-248.
- [16] 储颖,糜华,纪震,等.基于粒子群优化的快速细菌群游算法[J].数据采集与处理,2010,25(4):442-448.
- [17] KENNEDY J, EBERHART R C. Particle swarm optimization[C]. Proceedings of the IEEE International Conference on Neural Networks, Australia, 1995:1942-1948.
- [18] METROPOLIS N, ROSENBLUTH A W, ROSENBLUTH M N, *et al.* Equation of state calculation by fast computing machines[J]. Journal of Chemical Physics, 1953, 21(6):1087-1092.
- [19] BERMEJO E, CORDÁN, DAMAS S, *et al.* A comparative study on the application of advanced bacterial foraging models to image registration[J]. Information Sciences, 2015(1):160-181.
- [20] GUNEY K, BASBUG S. Bacterial foraging algorithm for null synthesizing of linear antenna arrays by controlling only the element positions[J]. J. Optim Theory Appl, 2011(1):635-656.
- [21] DEVI S, GEETHANJALI M. Application of modified bacterial foraging optimization algorithm for optimal placement and sizing of distributed Generation[J]. Expert Systems with Ap-

- plications,2014,41(1):2 772-2 781.
- [22] 姜建国,周佳薇,郑迎春,等.一种双种群细菌觅食优化算法[J].深圳大学学报:理工版,2014,31(1):43-51.
- [23] 章国勇,伍永刚,谭宇翔.一种具有量子行为的细菌觅食优化算法[J].电子与信息学报,2015,35(3):614-621.
- [24] BISWAS A,DASGUPTA S,DAS S, *et al.* A synergy of differential evolution and bacterial foraging algorithm for global optimization[J]. Neural Network World,2007,17(6):607-626.
- [25] 刘小龙,赵奎领.基于免疫算法的细菌觅食优化算法[J].计算机应用,2012,32(3):634-637.
- [26] 张伯明,陈寿孙,严正.高等电力网络分析[M].第2版.北京:清华大学出版社,2007:325-328.
- [27] 张燕,许伟伟.基于粒子群-差异进化混合算法的电力系统无功优化[J].电工电能新技术,2014,33(9):48-51.
- [28] WU Q H,CAO Y J,WEN J Y. Optimal reactive power dispatch using an adaptive genetic algorithm[J]. Int. J. Electr Power & Energy Syst. 1998,20(8):563-569.

(编辑 白林雪)

(上接第166页)

跳出局部极值点的能力.算法7是3种改进方案相结合的改进算法,随机选择变异策略综合了DE/rand/1和DE/best/1的优点,即增加了种群的多样性,又提高了搜索的快速性.由于算法7在部分代中采用了DE/rand/1变异策略,减少了计算变异率的次数,导致算法7的时耗及快速性都要优于算法6.

在跳出局部极值点能力、时耗、快速性3个指标中,最重要的指标是跳出局部极值点的能力,它是辨识成败与否的关键.因此,综合考虑,算法7的性能最优.

4 结 语

将3种改进设想,随机选择变异策略、自适应调整变异率和自适应调整交叉率组合应用于基本DE算法,提出了7种改进算法并应用于双容水箱液位模型闭环辨识案例中进行性能验证.对比基本DE算法辨识结果表明,7种算法跳出局部最优的能力都得到了提高;综合跳出局部最优能力、时耗、快速性3项性能表明,3种改进设想同时应用的改进算法性能最优.

参考文献:

- [1] 刘金琨,沈晓蓉,赵龙.系统辨识理论与MATLAB仿真

[M].北京:电子工业出版社,2013:32-58.

- [2] 裴振奎,刘真,赵艳丽.差分进化算法在多目标路径规划中的应用[J].辽宁工程技术大学学报:自然科学版,2010,29(5):899-902.
- [3] 郑慧涛,梅亚东,胡挺,等.改进差分进化算法在梯级水库优化调度中的应用[J].武汉大学学报:工学版,2013(1):57-61.
- [4] 常俊林,李亚朋,马小平,等.基于改进差分进化算法的PID优化设计[J].控制工程,2010,17(6):807-810.
- [5] 许津津,马进,唐永红,等.基于改进DE算法的负荷建模参数辨识[J].电力系统保护与控制,2009,37(24):36-40.
- [6] 薛晓岑,向文国,吕剑虹.基于差分进化与RBF神经网络的热工过程辨识[J].东南大学学报:自然科学版,2014,44(4):769-774.
- [7] 陶国正.基于改进微分进化算法的过程模型参数辨识[J].电气应用,2011(4):40-44.
- [8] 唐德翠,邓晓燕,朱学峰,等.改进差分进化算法辨识加药凝聚过程参数[J].控制工程,2010(4):80-82.
- [9] 任甜甜,张宏立.基于改进差分进化算法的非线性系统辨识[J].计算机仿真,2014,31(7):342-345.
- [10] 熊伟丽,陈敏芳,张乾,等.基于改进差分进化算法的非线性系统模型参数辨识[J].计算机应用研究,2014,31(1):124-127.
- [11] 张美春.差分进化算法与应用[M].北京:北京理工大学出版社,2014:10-11.
- [12] 张洪涛,胡红丽,徐欣航,等.基于粒子群算法的火电厂热工过程模型辨识[J].热力发电,2010,39(1):59-61.

(编辑 胡小萍)