

基于动态模式搜索的差分进化算法

代瑞瑞, 马永杰, 摆玉龙, 李 智

(西北师范大学 物理与电子工程学院, 兰州 730070)

摘 要: 针对传统差分进化算法在优化多维复杂函数时早熟收敛和收敛速度慢的问题, 提出一种基于模式搜索的差分进化算法。在优化过程中加入判断个体早熟收敛的机制, 若检测到有早熟现象, 以当前种群搜索到的最优解作为有效初始点进行模式搜索, 使算法跳出局部最优, 增强全局寻优能力。采用典型的测试函数进行仿真, 结果表明, 与基本差分进化算法和基于混沌搜索的差分进化算法相比, 该算法最易跳出局部最优解, 收敛精度较高, 优化性能较强。

关键词: 差分进化; 模式搜索; 函数优化; 全局寻优; 早熟收敛; 收敛精度

中文引用格式: 代瑞瑞, 马永杰, 摆玉龙, 等. 基于动态模式搜索的差分进化算法[J]. 计算机工程, 2016, 42(9): 163-167.

英文引用格式: Dai Ruirui, Ma Yongjie, Bai Yulong, et al. Differential Evolution Algorithm Based on Dynamic Pattern Search[J]. Computer Engineering, 2016, 42(9): 163-167.

Differential Evolution Algorithm Based on Dynamic Pattern Search

DAI Ruirui, MA Yongjie, BAI Yulong, LI Zhi

(College of Physics and Electronic Engineering, Northwest Normal University, Lanzhou 730070, China)

[Abstract] In order to avoid premature convergence and slow convergence of the complex function optimization in traditional Differential Evolution (DE) algorithms, an improved DE algorithm based on pattern search is presented. A mechanism to judge premature convergence is introduced. If premature convergence is detected, this algorithm will regard the optimal solution of the current population as an effective initial point for pattern search to avoid local optimum and strengthen the global optimization ability. Several typical test functions are used in simulation, and results show that the proposed algorithm has stronger ability to jump out of local optimal solution, higher convergence precision and stronger optimization performance compared with the basic DE algorithm and the DE algorithm based on chaos search.

[Key words] Differential Evolution (DE); pattern search; function optimization; global optimization; premature convergence; convergence precision

DOI: 10.3969/j.issn.1000-3428.2016.09.029

1 概述

差分进化 (Differential Evolution, DE) 算法是由 Storn 和 Price 提出的一种采用浮点实数编码的全局优化算法, 主要解决连续领域的优化问题^[1], 它根据种群间的差分信息对个体形成的扰动搜索空间进行变异、交叉和选择操作, 具有高可靠性、强鲁棒性以及良好的优化性能, 因此, 差分进化算法已经成为进化计算领域的研究热点。作为一种群智能优化算法, 差分进化算法和遗传算法、粒子群优化算法一样, 不能保证准确地收敛到全局最优^[2], 尤其是在优化多维复杂问题时, 差分进化算法的全局寻优能力

会急剧下降, 一旦陷入早熟收敛, 就会致使算法搜索停滞。针对算法存在的问题, 很多学者提出了相应改进的 DE 算法。

文献[3]提出模糊适应性 DE 算法, 其困难性在于模糊隶属函数的选取较为复杂。DE 算法性能的优劣很大程度上取决于其控制参数的选择^[4]; 文献[5]提出 SaDE (Self-adaptive Differential Evolution) 算法, 对缩放因子 F 和交叉率 CR 借鉴前期产生优质解的经验进行自适应调整, 在一定程度上能够获得比传统算法更好的性能; 文献[6]提出一种基于适应度值的自适应差分进化算法, 根据个体与最优个体适应度的相互作用自动调节缩放因子 F , 确保个体快速、

基金项目: 国家自然科学基金资助项目“进化计算类智能算法在数据同化误差处理中的应用研究”(41461078)。

作者简介: 代瑞瑞 (1990 -), 女, 硕士, 主研方向为进化算法; 马永杰 (通讯作者), 教授、博士; 摆玉龙, 副教授、博士; 李 智, 硕士。

收稿日期: 2015-07-20 **修回日期:** 2015-11-06 **E-mail:** rui9081rui@163.com

稳定地向最优值逼近;文献[7]提出一种将混沌遍历性和高斯局部搜索相结合的混合差分进化算法,提高了算法的收敛精度和全局寻优能力;文献[8]提出一种参数适应性分布式差分进化算法,对解空间进行广泛探索,取得了较好的优化性能;文献[9]提出一种群模式全局搜索算法,在模式搜索基础上引入群智能思想,具有较强的局部-全局搜索能力;文献[10]用随机动态局部搜索算子对当前最优蜜源进行局部搜索,提出一种改进的人工蜂群算法,加快了算法的收敛速度。

受文献[9-10]的启发,本文提出一种基于动态模式搜索的差分进化算法,在进行选择、交叉和变异操作后,判断算法是否陷入局部最优。如果陷入局部最优,则以当前群体最优解作为初始搜索点进行动态模式搜索^[11],促使算法向全局最优解逼近。

2 标准差分进化算法

DE 算法是一种基于种群的并行智能优化算法,首先种群个体以平均分布概率在搜索空间内随机初始化,然后根据父代个体间的差分向量进行变异、交叉和选择操作,直到达到最大迭代次数或满足精度要求为止。其中操作算子选择、交叉和变异3种基本形式,构成了该算法强大搜索能力的核心^[12]。

2.1 种群初始化

DE 算法采用实数编码,它首先在解的可搜索范围内随机产生初始种群:

$$\mathbf{x}_i(G) = [x_{i,1}, x_{i,2}, \dots, x_{i,D}], i = 1, 2, \dots, NP \quad (1)$$

$$x_{i,j} = a_j + \text{rand}(0,1) \cdot (b_j - a_j), j = 1, 2, \dots, D \quad (2)$$

其中, D 为问题的维数; G 为当前进化代数; NP 为种群规模; b_j 和 a_j 表示 $x_{i,j}$ 的上下限(解的取值范围); $\text{rand}(0,1)$ 为 $(0,1)$ 间的均匀随机数。

2.2 差分变异

在 DE 算法中,种群内的不同个体间进行矢量差运算,然后与种群内的其他相异个体进行矢量和运算生成变异向量。根据变异向量生成方式不同,可形成多种变异策略^[13],其中变异算子 DE/rand/1 是最常用的一类变异算子, rand 代表在目标群体中随机选择一个个体并在其附近进行搜索, 1 代表差分矢量的数目有一个, DE/rand/1 的方程描述为:

$$\mathbf{v}_i(G+1) = \mathbf{x}_{r_1}(G) + F \cdot (\mathbf{x}_{r_2}(G) - \mathbf{x}_{r_3}(G)) \quad (3)$$

其中, $r_1 \neq r_2 \neq r_3 \neq i$; F 为缩放因子。

2.3 交叉

交叉操作是将变异得到的变异个体 $\mathbf{v}_i(G+1)$ 与目标个体 $\mathbf{x}_i(G)$ 进行交叉生成实验向量 $\mathbf{u}_i(G+1)$ 的过程,DE 算法有 2 种交叉方式:二项式交叉(用 bin 表示)和指数交叉(用 exp 表示)。基本差分进化算

法一般采用二项式交叉方式产生候选个体,且在交叉后的实验向量中至少有一维来自变异向量 $\mathbf{v}_i(G+1)$,其操作方程如下:

$$\mathbf{u}_i(G+1) = \begin{cases} \mathbf{v}_i(G+1) & (\text{rand}(0,1) \leq CR) \cup (j = j_{\text{rand}}) \\ \mathbf{x}_i(G) & \text{其他} \end{cases} \quad (4)$$

其中, CR 是交叉率,在 $(0,1)$ 范围内取值; j_{rand} 的值在 $[1,D]$ 范围内随机选择,且为整数^[14]。

2.4 选择

差分进化算法采用“贪婪”选择策略,目标向量和实验向量根据目标函数适应值的大小进行竞争,较优的个体将会作为新生个体,具体操作如下:

$$\mathbf{x}_{i,j}(G+1) = \begin{cases} \mathbf{u}_i(G+1) & f(\mathbf{u}_i(G+1)) < f(\mathbf{x}_i(G)) \\ \mathbf{x}_i(G) & \text{其他} \end{cases} \quad (5)$$

3 群体适应度方差

随着种群的不断进化,种群多样性的快速下降容易造成种群早熟收敛,种群个体之间的差异越来越小,形成了一种“聚集”现象,这表现在种群个体适应度之间的差异越来越小^[13],下面给出群体适应度方差的定义具体描述种群的这种状态。设群体规模为 NP , f_i 为第 i 个个体的适应度, f_{avg} 为种群当前的平均适应度,群体适应度方差定义为^[15]:

$$\delta^2 = \sum_{i=1}^{NP} \left| \frac{f_i - f_{\text{avg}}}{f} \right|^2 \quad (6)$$

其中, f 为归一化定标因子,其作用是限制群体适应度方差 δ^2 的大小,其定义公式如下:

$$f = \begin{cases} \max\{|f_i - f_{\text{avg}}|\} & \max\{|f_i - f_{\text{avg}}|\} > 1 \\ 1 & \text{其他} \end{cases} \quad (7)$$

其中, $i = 1, 2, \dots, NP$ 。由式(6)可以看出, δ^2 可以显现种群所有个体的“聚集”程度, δ^2 越小,种群个体聚集程度越大^[16],个体差异越小,个体适应度差异越小,算法易早熟;反之,种群个体分散度越大,个体差异越大,搜索范围越广,算法越容易跳出局部最优。随着种群不断进化,种群个体适应度差异越来越小,故群体适应度方差 δ^2 会越来越小,当 δ^2 小到某一程度,即 $\delta^2 < c$ (c 为某一特定临界值)时,认为该算法已经陷入了局部最优。

4 基于模式搜索的差分进化算法

4.1 模式搜索算法

模式搜索法又称为 Hooke-Jeeves (HJ)^[14] 算法,是由 Hooke 和 Jeeves 提出的一种直接局部搜索算法,主要由探测移动和模式移动 2 种移动过程构成^[16],沿坐标轴方向上的寻优移动称为探测移动,

在2个相邻点连线方向上的移动成为模式移动^[17],具体移动过程为:探测移动在某个参考点附近沿坐标轴方向逐一进行探测,目的是探测到下降有利的方向,通过探测移动获得一个更优值的点(基点),从基点出发,沿参考点到基点的方向进行移动的过程即是模式移动^[18],2种移动模式交替进行,使算法顺着函数值下降的最有利方向逼近函数极小点。模式搜索的算法步骤如下:

步骤1 初始化:给定初始点 $x(0)$, n 个坐标方向 $e_1, e_2, \dots, e_n$, 初始步长 $d(0) = (d_1(0), d_2(0), \dots, d_n(0)) > 0$, 加速系数 $a \geq 0$, 缩减率 $b \in (0, 1)$, 精度 $\varepsilon > 0$ 。置 $y(0) = x(0)$, 迭代次数 $k = 1$ 。

步骤2 探测移动:对于 $i = 1, 2, \dots, n$, 如果 $f(y_i + d) < f(y_i)$, 置 $y_{i+1} = y_i + d$; 否则 $f(y_i - d) < f(y_i)$, 置 $y_{i+1} = y_i - d$; 否则, 置 $y_{i+1} = y_i$ 。

步骤3 模式移动:若 $f(y_{n+1}) < f(x_k)$, 置 $x_{k+1} = y_{n+1}$, $y_1 = x_{k+1}$, 增大步长 $d = a \times d$, $k = k + 1$, 转步骤2。

步骤4 终止判断:若 $d \leq \varepsilon$, 停止迭代, 得到当前最优解 x_k ; 否则, 置 $d = b \times d$, $k = k + 1$, 转步骤2。

4.2 算法描述

DE算法在优化多维复杂函数时出现的早熟现象以及收敛速度慢的问题,严重影响了算法的优化性能,因此,本文先利用DE算法进行全局搜索,随后判断个体是否陷入局部最优,如果算法搜索到了局部最优解,以当前种群搜索到的历史最优解作为初始点进行模式搜索,使算法跳出局部最优解,逐渐向全局最优解靠近。据以上分析,其流程描述如下。

步骤1 初始化:种群规模 NP , 缩放因子 F , 交叉率 CR , 最大迭代次数 $G_{\max}$, 输入变量的上下界 $X_{\min}$ 和 $X_{\max}$, 变量维数 D , 模式搜索最大迭代次数 $k_{\max}$, 初始步长 $d(0)$, 模式搜索精度 ε , 加速系数 a , 缩减率 b 。

步骤2 种群评价:对初始种群个体进行评价,找出适应度最佳个体 X_b , 并记录 X_b 的适应度 P_b 。

步骤3 变异操作:根据式(3)生成变异向量 $v_i(G+1)$ 。

步骤4 交叉操作:根据式(4)生成试验向量 $u_i(G+1)$ 。

步骤5 选择操作:根据式(5)得到下一代个体 $x_{i,j}(G+1)$ 。

步骤6 判断早熟:若适应度方差 $\delta^2 < c$, 转步骤7, 否则转步骤8。

步骤7 按照模式搜索算法对当前种群进行模式搜索。

步骤8 更新种群最优个体 X_b 及 X_b 的适应度 P_b 。

步骤9 判断是否达到最大迭代次数,若达到最大迭代次数,输出最优解,终止算法;否则转步骤2。

5 仿真与结果分析

本文仿真环境为 Matlab12a, Win7 系统, 主频 2.50 GHz, 内存 3.58 GB, 文中用到的所有函数都是最小化的, 测试函数分别为: f_1 (Sphere 函数) 为单模态函数, f_2 (Ackley 函数), f_3 (Griewank 函数), f_4 (Schaffer 函数) 和 f_5 (Rastrigin 函数) 为多模态函数, 这 5 个函数的全局最小值都是 0, 仿真测试中: 变量维数 $D = 30$, DE 算法的最大迭代次数 $G_{\max} = 1\,000$, 种群数目 $NP = 100$, 变异因子 $F = 0.6$, 交叉因子 $CR = 0.5$; 在模式搜索算子中: 初始步长 $d(0) = 0.02 \times \text{ones}(1, D)$, 精度 $\varepsilon = 1.0e-06$, 加速系数 $a = 1.5$, 减速系数 $b = 0.5$ ^[16], 模式搜索的最大迭代次数 $k_{\max}$ 取 150, 借鉴文献[12], 适应度方差判别阈值 c 取 0.001。

$$f_1(x) = \sum_{i=1}^D x_i^2, x_i \in [-100, 100], D = 30 \quad (8)$$

$$f_2(x) = -20 \exp \left(0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left[\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right] + 20 + e$$

$$x_i \in [-32, 32], D = 30 \quad (9)$$

$$f_3(x) = \frac{1}{4\,000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$$

$$x_i \in [-600, 600], D = 30 \quad (10)$$

$$f_4(x) = 0.5 - \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$$

$$x_i \in [-5.12, 5.12], D = 2 \quad (11)$$

$$f_5(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$$

$$x_i \in [-10, 10], D = 30 \quad (12)$$

相对于标准 DE 算法的静态化搜索策略, 虽然 HJDE 算法优化性能较好, 但其操作流程上的复杂程度较标准 DE 算法有所提高。分别用 DE 算法和 HJDE 算法对函数 $f_1 \sim f_5$ 进行优化迭代 1 000 次, 记录平均每迭代一次所用时间如表 1 所示, 可以看出, 对于函数 $f_1 \sim f_5$, HJDE 算法平均每迭代一次所用时间均比 DE 算法长。对于标准 DE 算法, 差分变异和交叉操作生成与其相对应的变异向量和实验向量, 然后将个体的目标向量和实验向量的适应度值进行比较, 经过贪婪选择操作确定的每一个个体所对应的子代个体, 最后更新种群最优个体和算法终止判断。与标准 DE 算法相比, HJDE 算法在操作步骤上又增加了早熟收敛判断机制和模式搜索机制, 即种群进行变异、交叉和选择基本操作后, 还要对当前种

群的适应度方差进行评估,对评估出的早熟停滞群体进行模式细搜索,然后再更新种群最优个体和算法终止判断,因此,HJDE 算法执行语句较 DE 算法多,算法执行时间较 DE 算法长,操作流程也较复杂。

表 1 DE 算法和 HJDE 算法的运行时间比较^s

测试函数	DE 算法	HJDE 算法
f_1	0.002 6	0.003 5
f_2	0.005 9	0.006 8
f_3	0.003 7	0.004 5
f_4	0.003 0	0.003 9
f_5	0.004 1	0.005 2

为了验证本文算法的可行性,将所提出的基于模式搜索的差分进化算法(HJDE)、基本差分进化算法(DE)以及混沌搜索的差分进化算法(CDE)进行算法性能比较,分别用基本差分进化算法(DE)、混沌搜索的差分进化算法(CDE)和模式搜索的差分进化算法(HJDE)对 $f_1 \sim f_5$ 函数进行优化,3 种算法对每一个测试函数独立运行 20 次,所得最优值的最小值 Min 、最大值 Max 、平均值 Fav 以及达到最优效果时的迭代次数 Nfe 的记录如表 2 所示, Min 、 Max 和 Fav 反映了解质量的好坏,迭代次数 Nfe 反映了算法收敛速度的快慢。

表 2 3 种算法测试结果比较($D=30$)

函数	参数	DE 算法	CDE 算法	HJDE 算法
f_1	Max	2.132 7e-04	1.014 9e-10	0.000 0
	Min	8.087 9e-05	5.972 8e-11	0.000 0
	Fav	1.358 3e-04	8.156 8e-11	0.000 0
	Nfe	1 000	1 000	749
f_2	Max	9.033 7e-03	7.614 8e-06	8.881 7e-16
	Min	5.187 3e-03	6.006 6e-06	8.881 7e-16
	Fav	6.337 3e-03	6.810 9e-06	8.881 7e-16
	Nfe	999	999	816
f_3	Max	2.124 3e-02	8.219 5e-02	8.219 5e-02
	Min	4.104 3e-04	4.791 3e-12	0.000 0
	Fav	3.335 3e-03	7.759 3e-03	6.217 4e-03
	Nfe	999	999	496
f_4	Max	0.000 0	0.000 0	0.000 0
	Min	0.000 0	0.000 0	0.000 0
	Fav	0.000 0	0.000 0	0.000 0
	Nfe	590	335	61
f_5	Max	1.633 2e+02	1.662 4e+02	1.451 5e+02
	Min	1.515 8e+02	1.410 7e+02	1.258 8e+02
	Fav	1.567 1e+02	1.549 3e+02	1.384 4e+02
	Nfe	657	956	642

由表 2 中的结果比较可以看出,对于优化函数 f_1 和 f_3 ,HJDE 算法在 1 000 代以内就能跳出局部最优解,而其他 2 种算法都没有搜索到全局最优解,且 CDE 算法的收敛精度要优于 DE 算法;对于优化函数 f_2 ,HJDE 的收敛精度最高,收敛速度最快,CDE 算法次之,DE 算法收敛精度最低;对于优化函数 f_4 ,3 种算法均搜索到了全局最优解,HJDE、CDE 和 DE 分别在第 61 代、第 335 代和第 590 代达到全局最优,HJDE 算法的收敛速度是最快的,CDE 算法次之,DE 算法最慢;对于优化函数 f_5 ,3 种算法中 HJDE 算法的收敛精度和速度优于其他 2 种算法。通过以上分析可以看出,HJDE 算法具有较高的收敛速度和收敛精度,是一种优秀的函数寻优算法。

为了更加直观地展示 3 种算法的寻优结果,给出 3 种算法对函数 $f_1 \sim f_5$ 的寻优曲线,如图 1~图 5 所示,其中横坐标为进化代数,纵坐标为 20 次函数寻优结果的均值。

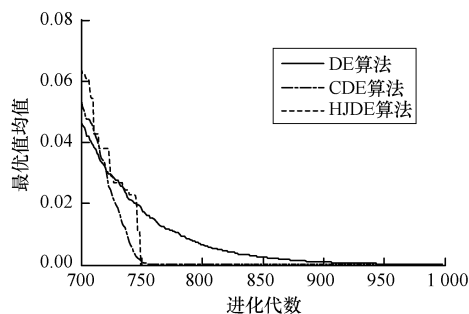


图 1 3 种算法对 f_1 的寻优曲线

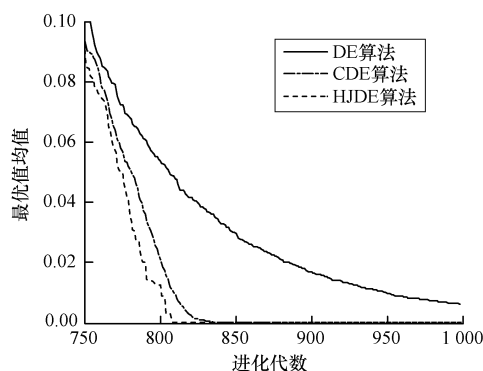


图 2 3 种算法对 f_2 的寻优曲线

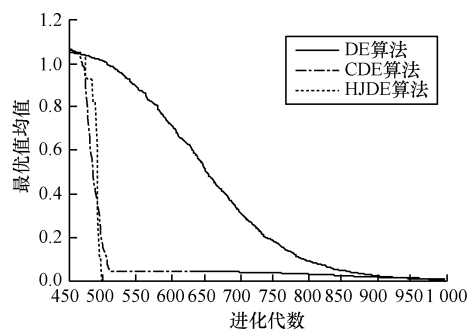
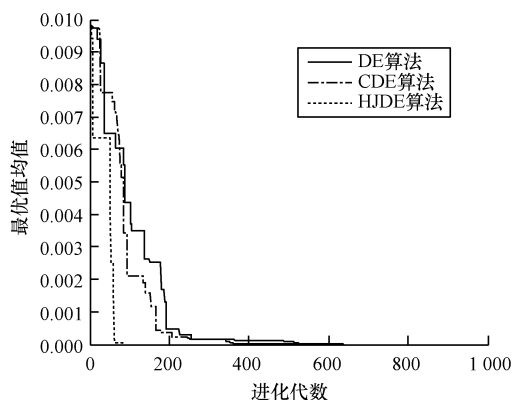
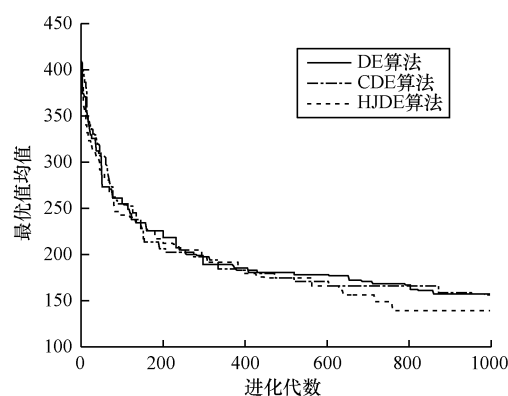


图 3 3 种算法对 f_3 的寻优曲线

图4 3种算法对 f_4 的寻优曲线图5 3种算法对 f_5 的寻优曲线

从5个寻优曲线图可以看出,不论是从收敛速度上还是收敛精度上,HJDE算法都表现出了较好的性能。

6 结束语

本文提出一种基于动态模式搜索的DE算法,首先利用基本DE算法进行全局搜索,然后根据个体收敛情况进行局部模式搜索。结合DE算法的全局搜索能力和模式搜索算法的强局部搜索能力,该算法能更快、更精确地逼近全局最优解。在标准测试函数上的仿真结果表明,HJDE算法能够弥补传统DE算法优化多维复杂函数的不足,具有较好的鲁棒性和稳定性,与其他2种算法相比,收敛性能更强。今后将利用该算法测试更多复杂测试函数的性能,并将其应用于优化实际问题中。

参考文献

- [1] 张春美,陈杰,辛斌.武器目标分配问题的离散差分进化算法[J].北京理工大学学报,2014,34(3):289-293,321.
- [2] 马永杰,摆玉龙,蒋兆远.快速约束多目标进化算法及其收敛性[J].系统工程理论与实践,2009,29(5):149-157.
- [3] Liu Junhong, Lampinen J. A Fuzzy Adaptive Differential Evolution Algorithm[J]. Soft Computing, 2005, 9(6): 448-462.
- [4] Mandal K K, Chakraborty N. Parameter Study of Differential Evolution Based Optimal Scheduling of Hydro-thermal Systems[J]. Journal of Hydro-environment Research, 2012, 7(1): 72-80.
- [5] Qin K, Huang V L, Suganthan P N. Differential Evolution Algorithm with Strategy Adaptation for Global Numerical Optimization[J]. IEEE Transactions on Evolutionary Computing, 2009, 13(2): 398-417.
- [6] 戈剑武,祁荣宾,钱锋,等.一种改进的自适应差分进化算法[J].华东理工大学学报:自然科学版,2009,35(4):600-605.
- [7] 贾东立,郑国苹.基于混沌和高斯局部优化的混合差分进化算法[J].控制与决策,2010,25(6):899-902.
- [8] 张春美,陈杰,辛斌.参数适应性分布式差分算法[J].控制与决策,2014,29(4):701-706.
- [9] 曲良东,何登旭,吴尽昭.一种群模式全局搜索算法[J].模式识别与人工智能,2013,26(6):592-597.
- [10] 刘三阳,张平,朱明敏.基于局部搜索的人工蜂群算法[J].控制与决策,2014,29(1):123-128.
- [11] 谢朝政,樊晓光,嵇真福.引入模式搜索算子的粒子群优化算法[J].微电子学与计算机,2015,32(2):97-99.
- [12] 马永杰,云文霞.遗传算法研究进展[J].计算机应用研究,2012,29(4):1201-1206.
- [13] 吴亮红.差分进化算法及应用研究[D].长沙:湖南大学,2007.
- [14] 张慧忠.基于双种群差分进化算法的输电网规划[J].山西科技,2014,29(3):62-64.
- [15] 刘军民,高岳林.基于混沌搜索的微分进化算法[J].计算机工程与应用,2009,44(12):66-68.
- [16] 邓泽喜,曹敦度,刘晓冀,等.一种新的差分进化算法[J].计算机工程与应用,2009,44(24):40-42.
- [17] 李立刚,张朝晖,戴永寿,等.基于改进模式搜索算法的天然气管网运行优化[J].中国石油大学学报:自然科学版,2012,36(4):139-143.
- [18] 柳贺,黄猛,柳桂国,等.基于混沌搜索和模式搜索的混合优化方法[J].华东理工大学学报:自然科学版,2008,34(1):126-130.

编辑 金胡考