

基于极值优化的混合差分进化算法

王丛佼 王锡淮 肖建梅

(上海海事大学电气自动化系 上海 201306)

摘要 针对标准差分进化算法在求解复杂优化问题时易陷入局部最优的问题,提出了一种基于极值动力学机制的混合差分进化算法。该算法的核心在于,当种群聚集度较高时,利用极值优化算法强大的波动性,通过引入基于种群的极值优化算法来提高种群多样性,从而协助差分进化算法跳出局部最优。仿真实验表明,该混合算法具有较好的全局收敛性,能有效避免早熟收敛。

关键词 差分进化,极值优化,混合算法,全局优化

中图分类号 TP18 **文献标识码** A

Hybrid Differential Evolutionary Algorithm Based on Extremal Optimization

WANG Cong-jiao WANG Xi-huai XIAO Jian-mei

(Department of Electrical Engineering and Automation, Shanghai Maritime University, Shanghai 201306, China)

Abstract A new hybrid algorithm based on differential evolution (DE) and extremal optimization (EO) was proposed to solve the premature convergence and low precision of standard differential evolution when it is applied to complex optimization problems. The key points of it lie in: the hybrid algorithm introduces the population-based extremal optimization algorithm in the iteration process of DE when population aggregation gets the high degree, which uses the volatility of EO to increase the diversity of population and the ability of breaking away from the local optimum. Simulations show that the hybrid algorithm has remarkable global convergence ability, and can avoid the premature convergence effectively.

Keywords Differential evolution, Extremal optimization, Hybrid algorithm, Global optimization

1 引言

差分进化(Differential Evolution, DE)算法是由 Rainer Storn 和 Kenneth Price 提出的一种基于群体差异的启发式优化算法^[1]。DE 算法通过群体内个体间的合作与竞争来实现对优化问题的求解,具有受控参数少、操作算子简单和鲁棒性好等优点。作为一种简单实用的全局搜索算法,DE 算法已被证明是求解非线性、不可微和超高维等复杂函数的有效手段^[2],在科学研究与实际工程领域都得到了广泛的应用^[3,4]。但与其他进化算法(如遗传算法、粒子群算法等)一样,标准 DE 算法在进化后期也存在早熟收敛现象,尤其对于复杂多峰函数的优化,其全局收敛效率受到限制^[5,6]。为解决此问题,许多学者在控制参数设置^[7]、变异算子改进^[8]和与其他寻优算法相结合^[9]等方面做了相关改进研究。

由 Boettcher 和 Percus 提出的极值优化(Extremal Optimization, EO)算法^[10]是一种新开发的基于局部搜索的启发式算法。与其他优化算法不同,该算法模拟远离平衡态系统,不会收敛到一个平衡态而出现断续平衡,由此产生的波动性使其具有更好的持续搜索和跳出局部最优的能力。本文首先对基本 EO 算法进行了改进,同时结合 DE 算法强大的全局寻优能力和 EO 算法精细的局部寻优能力设计了一种新的混

合算法(DE-EO):整个算法采用 DE 算法的框架,通过差分进化操作提高算法的收敛速度和搜索性能;当判定群体适应值方差小于一定阈值时引入改进后的 EO 算法,通过更新个体的较差或最差组元来提高种群多样性,协助 DE 算法跳出局部最优。研究表明:该混合算法能有效避免早熟收敛现象,具有良好的全局收敛性。

2 差分进化算法

DE 算法是采用实数编码的进化算法,其原理和结构类似于遗传算法,同样采用遗传操作(变异、杂交和选择)。其与遗传算法不同的是:1)在变异操作方面使用差分策略,有效利用了群体分布信息;2)在选择操作方面采用一对一竞争选择策略,降低了遗传操作的复杂性。

对于一般的全局优化问题:

$$\begin{aligned} \min f(x) \\ \text{s. t. } x \in X \end{aligned}$$

式中, $x \in R^n$ 是决策变量, $f(x)$ 为目标函数, $X \subset R^n$ 为约束集或可行域。

DE 算法的基本流程为:

1)种群初始化。设定种群规模为 NP ,每个个体自变量有 D 维。种群中第 i 个个体 x_{ij} ($i=1, \dots, NP; j=1, \dots, D$) 为

到稿日期:2012-11-05 返修日期:2013-03-18 本文受上海市教科委科研创新重点项目(12ZZ158),上海市教委重点学科建设项目(J50602)资助。

王丛佼(1984-),女,博士生,主要研究方向为智能控制、智能信息处理等,E-mail: qinxiaojiaji@hotmai. com;王锡淮(1961-),男,教授,博士生导师,主要研究方向为智能控制、智能信息处理、智能图像处理等;肖建梅(1962-),女,教授,主要研究方向为智能控制、智能信息处理等。

一实数在其自变量范围 $[l, h_j]$ 内随机均匀初始化。

2) 变异操作。DE 算法利用种群中不同个体间的差分向量对个体进行扰动,产生变异向量 u_i^{t+1} 。基本 DE 算法有多种不同的变异方式,通常以 $DE/a/b$ 来表示,其中 a 代表基向量的选择方式, b 表示差分向量的个数。一般性常用的变异算子有:

$$\text{DE/rand/1:} \\ u_i^{t+1} = x_{r_1} + F(x_{r_2} - x_{r_3}) \quad (1)$$

$$\text{DE/best/1:} \\ u_i^{t+1} = x_{\text{best}} + F(x_{r_2} - x_{r_3}) \quad (2)$$

式中, $r_1, r_2, r_3 \in [1, 2, \dots, NP]$,为互不相同的整数; t 为当前迭代次数; x_{best} 为第 t 次迭代的当前最优解; F 为缩放因子, $F \in (0, 2)$ 。

3) 交叉操作。DE 算法的交叉算子分为二项式交叉和指数交叉。在二项式交叉中,首先通过随机选择,使得试验向量 u_i^{t+1} 至少有一维由变异向量 u_i^{t+1} 贡献;对于其它维,由交叉概率因子 CR 决定由 u_i^{t+1} 贡献还是由 x_i 贡献, $CR \in [0, 1]$ 。

$$u_{ij}^{t+1} = \begin{cases} u_{ij}^{t+1}, \text{rand}(j) \leq CR \text{ or } j = \text{randn}(i) \\ x_{ij}, \text{rand}(j) > CR \text{ and } j \neq \text{randn}(i) \end{cases} \quad (3)$$

式中, $\text{rand}(j) \in [0, 1]$ 为均匀分布的随机数; $\text{randn}(i) \in [1, 2, \dots, D]$ 为随机选择的维数索引。

4) 选择操作。DE 算法采用贪婪的选择策略,即只有目标函数值更优的个体进入到下一代。

$$x_i^{t+1} = \begin{cases} u_i^{t+1}, f(u_i^{t+1}) < f(x_i^{t+1}) \\ x_i^t, f(u_i^{t+1}) \geq f(x_i^{t+1}) \end{cases} \quad (4)$$

5) 终止条件。如果迭代次数 G 达到最大迭代次数 G_{max} 或者最优个体目标函数值的精度达到要求,则算法停止搜索;否则,种群继续执行变异、交叉和选择操作。本文选择达到最大迭代次数作为终止条件。

3 基于极值优化的混合差分进化算法

3.1 极值优化算法及改进

EO 算法是受复杂系统自组织临界思想和 Bak-Sneppen 生物演化模型启发而来的一种自组织优化算法^[11]。该算法的核心思想是,不断以新的随机组元无条件替换个体中适应度最差或较差的组元,使得个体的所有组元协同进化,最后达到近似最优解或最优解,实现寻优。这种进化模式也被称为极值动力学机制。由于其内在的极值动力学机制,EO 算法在进化后期仍具有很大的波动性,能够持续搜索,从而避免算法早熟收敛。

基本 EO 算法在进化过程中只有一个个体进行演化,为了其能够与基于群体的 DE 算法结合,帮助算法跳出局部最优,本文将种群的概念引入 EO 算法中,提出适合求解数值优化问题的基于种群的 EO 算法。

EO 算法只有变异操作,没有交叉操作,因此其变异操作对算法的性能具有重要的作用。单个个体的 EO 算法采用随机生成新组元的变异方式更新最差组元;而对于种群的 EO 算法,为更好地定义每个个体的较差或最差组元,并使用具有某种指引性的更新机制来替代无序的随机更新方式,改进的 EO 算法需要引入新的变异算子。柯西算子和高斯算子是两大常用的变异算子。文献[12]通过对两大算子的深入研究,指出柯西变异具有较好的全局搜索能力,与高斯变异相比具

有更大的搜索步长,可以使搜索点快速向全局最优优点靠拢,因此容易帮助算法跳出局部极值点。基于此优点,种群的 EO 算法采用柯西变异作为变异算子。

$$x_{ij}^* = x_{ij} + \delta_k \quad (5)$$

式中, x_{ij}^* 和 x_{ij} 分别表示第 i 个个体的第 j 维组元在变异前和变异后的值; δ_k 表示柯西随机数。若进行变异后,新个体超出自变量范围,则直接取其上界或下界作为新的变量。

除了变异操作,定义合适的适应度函数也是 EO 算法的关键环节。事先确定的适应度函数能够引导个体进化的方向,加速个体的搜索。因此,对于无约束的全局优化问题,每个组元的适应度 λ_{ij} 将定义为其变异后对群体目标函数值的贡献大小,适应度函数表达式如下:

$$\lambda_{ij} = f(X_{ij}^*) - f(X_{\text{best}}) \quad (6)$$

式中, X_{ij}^* 是只对 x_{ij} 变异而保持其他组元不变时得到的新个体; $f(X_{ij}^*)$ 是 X_{ij}^* 的目标函数值; $f(X_{\text{best}})$ 是当前最优解 X_{best} 的目标函数值。可见,组元的适应度 λ_{ij} 越好,其发生变异的概率就越低。

改进的 EO 算法具体流程见图 1。

1. 设定种群个体数量 n 和维数 d ;随机生成初始种群 $(X_1, \dots, X_n)$, $i \in [1, \dots, n]$;
2. 计算每个个体的目标函数值 $f(X_i)$,选择目标函数值最小的个体为当前最优解 X_{best} ,其对应的目标函数值为 $f(X_{\text{best}})$;
3. 对种群中每一个个体 X_i 进行如下操作:
 - 1) 采用式(5)对个体中每个组元 x_{ij} , $j \in [1, \dots, d]$ 进行变异操作,同时保持其他组元不变,得到 d 个个体 X_{ij}^* ;
 - 2) 根据式(6)计算适应度 λ_{ij} 并对其排序,找出适应度最小的子个体 X_{iq}^* , $q \in [1, \dots, d]$,其对应的 x_{iq} 即为最差组元;
 - 3) 令 $X_i = X_{iq}^*$, $f(X_i) = f(X_{iq}^*)$;
 - 4) 如果 $f(X_i) < f(X_{\text{best}})$,则更新当前最优解,即令 $X_{\text{best}} = X_i$, $f(X_{\text{best}}) = f(X_i)$;否则,继续下一步;
4. 如果满足算法结束条件,则输出最优解 X_{best} 及其目标函数值 $f(X_{\text{best}})$;否则,返回步骤 3。

图 1 基于种群的 EO 算法的伪代码

3.2 混合 DE-EO 算法的思想

基本 DE 算法采用贪婪的选择方式,这种方式有利于提高收敛速度,但在算法进化后期,选择压力导致种群内个体很快趋同,从而大幅损失了种群多样性,增加了早熟收敛的风险。尤其对于求解复杂多峰函数的全局解,DE 算法很容易陷入局部最优而失效。为此,本文提出了一种基于极值优化的混合差分进化算法(DE-EO),一方面利用 DE 算法较好的搜索能力和收敛速度,可快速地逼近近似最优解;另一方面利用 EO 算法持续的波动性,可维护种群多样性,协助 DE 算法跳出局部最优以弥补其不足。

考虑到如果每次迭代都引入 EO 算法会大大增加计算成本,同时可能从一开始就破坏正确的进化方向,降低原本 DE 算法的收敛速度,因此本文提出当发现 DE 算法倾向于局部收敛时,再引入 3.1 节提出的改进 EO 算法。这样,混合 DE-EO 算法能够保持快速的收敛速度,同时在 DE 算法陷入局部最优时,利用 EO 算法对个体中的最差组元进行变异以产生新的个体,不管新个体的适应度是否优于原个体,都用新个体取代原个体,从而增加个体的多变性,开拓对局部空间的持续搜索,起到了协调选择压力与种群多样性之间矛盾的作用。

本文使用群体适应值方差 $\sigma^{[13]}$ 来度量种群多样性,判定

早熟现象。设种群规模为 NP , f_i 为第 i 个个体的适应值, f_{avg} 为种群平均适应值, 则群体适应值方差定义为:

$$\theta = \frac{1}{NP} \sum_{i=1}^{NP} \left| \frac{f_i - f_{avg}}{\max\{|f_i - f_{avg}|\}} \right|^2 \quad (7)$$

由式(7)可知, θ 反映了种群中所有个体的聚集程度。显然, θ 越小, 个体聚集度越高。个体的高度聚集一方面导致种群多样性的大幅降低, 个体之间的差分向量越来越小, 阻碍算法的进一步搜索, 一旦早熟收敛便无法跳出局部最优; 另一方面, 由式(2)可见, 若变异个体 x_{best} 为一局部极小点, 则由于其余个体都聚集在其周围, 算法出现早熟, 并且即使选用其他变异方式, 变异个体为种群中任意别的个体, 若某一个体是局部极小点, 其也同样有陷入局部最优的可能, 算法的收敛速度也会变慢。因此设定群体适应值方差阈值 ϵ , 当最优解的适应值未达到精度要求且 $\theta < \epsilon$ 时, 说明算法还未收敛到全局最优且极可能发生局部收敛, 此时进行 EO 算法操作, 在原来个体一定范围的区域内产生新个体并更新当前最优解, 从而打散个体分布, 降低个体的聚集度。

3.3 混合 DE-EO 算法的实现

混合 DE-EO 算法的具体步骤如图 2 所示。

1. 设定种群规模 NP 、变量维数 D 、算法参数 F 、 CR 以及群体适应值方差阈值 ϵ ;

2. 指定最大迭代次数 G_{max} , 令迭代次数 $G=0$; 初始化种群;
3. 评价每个个体的目标函数值, 并更新当前最优解和全局最优解;
4. 分别根据式(1)、式(3)、式(4)进行 DE 的变异、交叉和选择操作;
5. 评价新产生个体的目标函数值, 并更新当前最优解和全局最优解;
6. 如果满足终止条件, 则算法停止, 输出全局最优解及其目标函数值; 否则, 继续下一步;
7. 根据式(7)计算当前群体适应值方差;
8. 如果 $\theta < \epsilon$, 则转向图 1 所示的 EO 子步骤, 否则, 令 $G=G+1$, 转向步骤 4。

图 2 混合 DE-EO 算法的伪代码

4 算法测试与分析

4.1 测试函数与实验设置

为验证 DE-EO 算法的有效性, 本文选择 5 个基准 Benchmark 函数进行测试。表 1 列出了这 5 个 Benchmark 函数的具体表达形式、变量定义域和理论目标函数最优值。其中, D 表示变量的维数, 本文分别测试了函数为 25 维和 100 维时算法的性能。所有函数除了 f_1 外, 均为复杂多峰函数, 在定义域内存在多个局部极小点, 因此可有效检验算法全局搜索和跳出局部最优解的能力。

表 1 Benchmark 函数列表

函数名称	函数表达式	定义域	最优位置	最优值
Sphere	$f_1(x) = \sum_{i=1}^D x_i^2$	$[-100, 100]^D$	$(0, 0, \dots, 0)$	0
Rosenbrock	$f_2(x) = \sum_{i=1}^{D-1} (100 \cdot (x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	$[-30, 30]^D$	$(1, 1, \dots, 1)$	0
Rastrigin	$f_3(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$[-5.12, 5.12]^D$	$(0, 0, \dots, 0)$	0
Griewank	$f_4(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^D$	$(0, 0, \dots, 0)$	0
Ackley	$f_5(x) = 20 + e - 20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right)$	$[-32, 32]^D$	$(0, 0, \dots, 0)$	0

本文将 DE-EO 算法与标准 DE 算法进行了比较, 并分别选用两种变异策略: 1) rand/1/bin; 2) best/1/bin, 这两种策略是 DE 算法中最基本的变异形式, 具有良好的代表性, 其他变异形式均可视为在它们的基础上变化而得。为使得测试结果公平和合理, 所有算法对每个函数都独立运行 30 次, 使用相同的终止条件, 统一设置共同部分的参数。实验参数设置如下: 种群规模 $NP=10D$, 缩放因子 $F=0.5$, 交叉概率因子 $CR=0.9$; 群体适应值方差阈值 $\epsilon=0.001$; 25 维和 100 维函数的最大迭代次数分别为 2000 和 5000。

4.2 实验结果分析

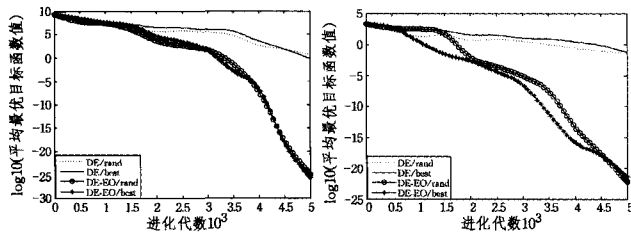
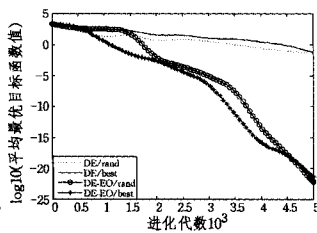
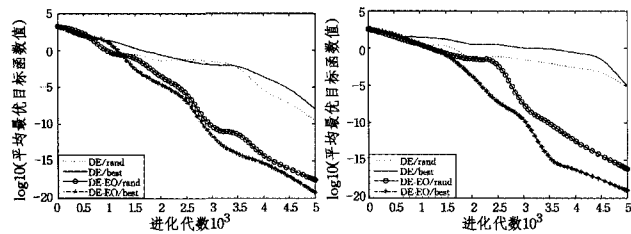
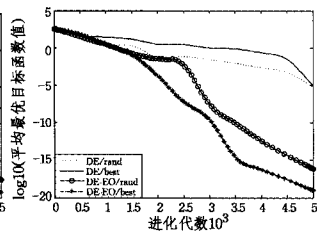
上述算法对 f_1-f_5 函数运行 30 次后得到的最优解平均值和标准差如表 2 所列。由表 2 可见, 在相同的参数条件下, DE-EO 算法的收敛精度高, 稳定性好。DE 算法虽然在单峰函数表现稳定, 但在有多个局部极值点的多峰函数上很容易产生早熟现象, 无法收敛到全局最优解; 反之, 本文提出的 DE-EO 算法在所有函数的求解中都表现出理想的寻优能力。其中, 对于单峰 f_1 函数, DE-EO 算法的测试结果的优势并不明显, 特别当函数为 100 维时, 改进作用甚微, 这是因为 DE 算法在求解单峰函数问题上本身具有良好的性能; 但对于多峰 f_2-f_5 函数, 无论函数是 25 维或是 100 维, DE-EO 算法的优化结果均明显优于标准 DE 算法。同时, 通过比较两种

变异策略可以看出, DE/rand/1 在大多数函数的求解上比 DE/best/1 表现更好, 但经过引入 EO 算法的改进后, DE-EO/best/1 在求解函数 f_1 、 f_4 和 f_5 上性能要好于 DE-EO/rand/1, 原因在于 DE/best/1 采用当前最优解 x_{best} 作为基向量, 理论收敛速度加快, 但在实际运行中, 一旦 x_{best} 为局部极值点, 则所有个体迅速向其靠拢而陷入局部最优, 阻碍算法的进一步搜索, 而改进后的 DE-EO/best/1 由于能够在发现算法倾向早熟时, 重新分布个体, 指引个体放弃局部极值点并向新的最优解收敛, 因此能够有效地进行持续搜索。

图 3—图 6 是在多峰 f_2-f_5 函数 ($D=100$) 上算法运行 30 次后得到的收敛过程曲线。为方便显示, 图中的纵坐标都采用目标函数值以 10 为底数的对数表示。图 3—图 6 显示, DE-EO 算法的收敛速度均高于 DE 算法。这主要是因为 DE-EO 算法能够一直较好地保持种群多样性。从图中可以看出, 在迭代前期, DE-EO 算法和 DE 算法的收敛特性相似, 但随着迭代次数的增加, DE 算法的种群多样性不断下降, 群体适应值方差变小, DE 算法产生早熟, 而 DE-EO 算法由于有效利用了极值优化的波动性, 因此能够帮助算法跳出局部最优, 持续对新空间进行搜索。以上结果表明, 基于极值优化的混合 DE-EO 算法能有效避免早熟, 尤其在多峰函数的优化问题中显示出更好的性能优势。

表2 4种算法30次独立运行的平均最优解及标准差

函数	维数	进化代数	平均最优解(标准差)			
			DE/rand/1	DE/best/1	DE-EO/rand/1	DE-EO/best/1
f_1	25	2000	4.1237e-29 (1.7374e-25)	9.1235e-27 (2.3574e-24)	3.5732e-32 (1.9765e-30)	2.8375e-34 (8.2367e-35)
	100	5000	6.4823e-24 (5.6873e-23)	8.3664e-24 (4.3875e-24)	9.3875e-25 (2.8375e-24)	4.8375e-25 (7.2874e-24)
f_2	25	2000	4.7382e-1 (4.3764e-3)	6.7356e-1 (2.8721e-2)	3.8573e-29 (5.2736e-28)	5.2837e-28 (6.2873e-26)
	100	5000	1.8312e+1 (1.8743e-1)	4.9652e+0 (1.7289e-2)	9.8277e-25 (5.8234e-24)	1.8745e-25 (2.8563e-23)
f_3	25	2000	1.3573e-3 (7.9356e-2)	5.7834e-2 (1.3672e-2)	4.7364e-25 (1.7463e-26)	2.8375e-23 (8.3753e-22)
	100	5000	3.5174e-2 (3.6312e-3)	5.4789e-2 (1.2587e-3)	5.8745e-22 (6.8723e-23)	3.7834e-21 (2.9876e-22)
f_4	25	2000	2.7364e-22 (9.7532e-24)	1.8698e-21 (3.9853e-22)	5.7365e-29 (2.8790e-29)	4.8576e-29 (2.3875e-30)
	100	5000	3.2831e-10 (7.1938e-11)	1.2547e-8 (2.9675e-9)	5.8746e-17 (1.8963e-17)	2.8735e-19 (1.0678e-18)
f_5	25	2000	5.4098e-8 (2.8714e-8)	1.2617e-8 (3.2974e-8)	5.8273e-22 (1.2384e-23)	2.3874e-23 (4.8273e-24)
	100	5000	7.6687e-5 (6.767e-5)	1.0245e-4 (4.8963e-5)	7.2613e-16 (5.3762e-17)	9.8356e-19 (2.0912e-20)

图3 f_2 函数测试结果图4 f_3 函数测试结果图5 f_4 函数测试结果图6 f_5 函数测试结果

结束语 针对标准 DE 算法容易出现早熟收敛现象,本文提出一种基于极值优化的混合 DE-EO 算法。该算法一方面充分发挥 DE 算法全局搜索和收敛速度快的优点,保证了算法的收敛性和稳定性;另一方面,利用 EO 算法持续的波动性,协助 DE 算法跳出局部最优,进一步增强了算法的寻优能力。测试表明,对于求解复杂多峰函数的优化问题,DE-EO 算法的寻优性能有明显提高。

参考文献

- [1] Storn R, Price K. Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces [J]. Journal of Global Optimization, 1997, 11(4): 341-359
- [2] Price K, Storn R. Differential Evolution-A Practical Approach to Global Optimization [M]. Berlin, Germany: Springer-Verlag, 2006: 133-152
- [3] Varadarajan M, Swarup K S. Network loss minimization with voltage security using differential evolution [J]. Electric Power Systems Research, 2008, 78(5): 815-823
- [4] Das S, Abraham A, Konar A. Automatic clustering using an improved differential evolution algorithm [J]. IEEE Transaction on Systems, Man and Cybernetics, 2008, 38(1): 218-236
- [5] Das S, Abraham A. Differential evolution using a neighborhood-based mutation operator [J]. IEEE Trans on Evolutionary Computation, 2009, 13(3): 526-553
- [6] 袁俊刚, 孙治国, 曲广吉. 差异演化算法的数值模拟研究 [J]. 系统仿真学报, 2007, 19(20): 4646-4648
- [7] Brest J, Greener S, Boskovic B. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems [J]. IEEE Transactions on Evolutionary Computation, 2006, 10(6): 646-657
- [8] 刘荣辉, 郑建国. 分区交叉差分进化算法及其约束优化 [J]. 计算机学报, 2012, 39(2): 283-287
- [9] Das S, Abraham A, Konar A. Particle swarm optimization and differential evolution algorithms: technical analysis, applications and hybridization perspectives [J]. Studies in Computational Intelligence, 2008, 116: 1-38
- [10] Boettcher S, Percus A G. Extremal Optimization: Methods Derived from Co-Evolution [A] // Proceedings of the Genetic and Evolutionary Computation Conference [C]. San Francisco: Morgan Kaufmann, 1999: 825-832
- [11] 齐洁, 汪定伟. 极值优化算法综述 [J]. 控制与决策, 2007, 22(10): 1081-1085
- [12] Lee C Y, Yao X. Evolutionary Algorithms with Adaptive Levy Mutations [C] // Proceedings of the 2001 Congress on Evolutionary Computation. 2001: 568-575
- [13] 吕振肃, 侯志荣. 自适应变异的粒子群优化算法 [J]. 电子学报, 2004, 32(3): 416-420