文章编号: 1000−8152(2008)05−0862−05

求解高维多模优化问题的自适应差分进化算法

张贵军, 王信波, 俞 立, 冯远静

(浙江工业大学 信息工程学院, 杭州 浙江 310032)

摘要: 在基变量选择方差理论分析的基础上, 提出一种自适应差分进化算法(ADE). ADE算法通过设计自适应收敛因子构建自调整的权重质心变异策略, 同时在交叉策略中引入发射、收缩两种单纯形操作算子, 保证算法全局搜索能力的同时, 能有效提高算法后期的局部增强能力. 30个优化问题的数值研究结果表明ADE算法具有比DE、DERL以及DERB三种算法更快的收敛速度和可靠性, 尤其适合于高维多模优化问题的求解.

关键词: 多模优化; 差分进化; 选择方差; 数值计算

中图分类号: TP391 **文献标识码**: A

Adaptive differential evolution for high-dimension multimodal optimization problems

ZHANG Gui-jun, WANG Xin-bo, YU Li, FENG Yuan-jing

(College of Information Engineering, Zhejiang University of Technology, Hangzhou Zhejiang 310032, China)

Abstract: Based on the theoretcal analysis of selective variance in mutation operator of original differential evolution (DE) algorithm, we proposed an adaptive differential evolution (ADE) algorithm to tackle the high-dimension multimodal optimization problems. In order to make a good tradeoff between the exploration and exploitation, ADE algorithm adopts an adaptive weighted centroid mutation strategy. Furthermore, modifications in mutation and crossover rule are suggested to the original DE algorithm to intensify the search around the global minima. These modifications intend to exploit the information derived from the previous function evaluations to improve the efficiency of the algorithm in the local search, without deteriorating the behavior of the original DE algorithm in the global search. Numerical experiments indicate that the resulting algorithm is considerably better and more efficient than the DE, DERL and DERB algorithms. Finally, a numerical study is carried out using a set of 30 test problems, many of which are inspired by practical applications.

Key words: multimodal optimization; differential evolution; selective variance; numerical experiment

1 引言(Introduction)

在实际工程应用中, 经常会遇到一类复杂的多模(多极值)连续函数全局优化问题:

$$\min \quad f(\boldsymbol{x}), \boldsymbol{x} \subseteq \Omega \subset \mathbb{R}^n. \tag{1}$$

其中: $\boldsymbol{x}$为可行域Ω上的n维连续(但不一定光滑)向量, f是定义在$\Omega \subset \mathbb{R}^n$上的非凸目标函数, 在限定的解向量空间可能存在多个全局最优解和大量的局部最优解. 基于梯度的传统优化方法[1]、Nelder-Mead[2]及Hooke-Jeeves[3]等算法本质上都属于一类局部搜索方法, 解的质量直接取决于起始点的选择, 对于复杂的高维多模优化问题, 由于其形成的超曲面极其粗糙, 这些方法基本上不可能得到问题的全局最优解. 确定性全局优化方法[4]通常依赖待解问题的先验知识, 如Lipshitz常数; 此外, 极高的计算复杂度也阻碍了其在高维问题中的应用. 正是在这种情况下, 随机性全局优化算法得到了广泛的应用.

近年来, 差分进化(DE)作为一种性能卓越的随机性算法在全局优化领域受到了广泛的关注[5]. Ali研究结果表明[6], DE算法比遗传算法(GA)[7]、控制随机搜索(CRS)[8]等算法具有更好的全局搜索和局部增强能力. 但是, Kaelo[9]的研究同时也表明在DE算法进化的后期(尤其是在最优解附近), 算法同样会出现收敛速度减慢的问题, 为此, Ali、Kaelo等学者相继提出了一些改进版本(如DEPD、TDEPD、DERL、DERB等算法)以改善原算法后期较弱的收敛能力[9,10]. 对于一些简单的低维测试函数, 这些算法取得了较好的效果, 但是对

收稿日期: 2006−10−12; 收修改稿日期: 2007−12−28.

基金项目: 国家863计划资助项目(2006AA04Z178); 浙江省科技论文重点项目(2008C23040).

于中等或大规模的高维优化问题, 收敛速度仍然是算法的瓶颈所在, 而且也极容易陷于局部最优解, 出现早熟对象.

本文在基本差分进化算法的基础上, 提出了一种适于求解高维多模优化问题的自适应差分进化算法(adaptive difference evolution, ADE). 通过理论分析及30个全局优化测试问题的求解验证了ADE算法的有效性, 同时还给出了ADE算法与DE、DERL以及DERB算法的比较结果.

2 理论分析(Theory analysis)

变异过程中基变量选择方案是影响算法性能的主要因素[9]. DE算法采用基变量均匀随机选择策略[5], 而DERL算法采用基变量联赛选择策略[9]. 为理解两种基变量选择策略对DE算法收敛性能影响, 以下分析过程中均假定: 算法不执行变异策略中增益操作、不执行交叉操作及执行全部替换操作.

2.1 均匀选择策略(Uniform selection strategy)

引理 1 设$X_0=\{x_1,x_2,\cdots,x_N\}$为初始群体, $F_0\equiv F(X_0)=\{f_1,f_2,\cdots,f_N\}$为初始群体适应度, $\mathrm{D}(F_0)$为初始群体适应度分布方差, N为群体规模. 如果采用均匀选择算子, 则第t代群体适应度分布平均方差[11]

$$\mathrm{E}(\mathrm{D}(F_t))=\left(1-\frac{\mathrm{D}(\bar{n})}{N-1}\right)^t\mathrm{D}(F_0), \tag{2}$$

其中随机变量$\bar{n}$表示群体中某个个体选择次数.

显然, $\mathrm{E}(\mathrm{D}(F_t))$与群体初始分布无关. 对于基本DE算法, $\bar{n}$服从二项式分布$\bar{n}\sim b(N,1/N)$, 即

$$\mathrm{D}(\bar{n})=1-\frac{1}{N}, \tag{3}$$

代入式(2)得

$$\mathrm{E}(\mathrm{D}(F_t))=\left(1-\frac{1}{N}\right)^t\mathrm{D}(F_0). \tag{4}$$

假设$\mathrm{E}(\mathrm{D}(F_t))<\epsilon$($\epsilon$为任意小正数)时算法终止, 则终止代数$T$可表示为

$$T=\frac{\ln(\frac{\epsilon}{\mathrm{D}(F_0)})}{\ln(1-\frac{1}{N})}. \tag{5}$$

考虑典型参数配置: $N=100$, $\mathrm{D}(F_0)=100$, $\epsilon=0.001$, 则$T>1000$. 这说明基变量均匀选择方案在有效增加全局搜索性能的情况下, 同时也牺牲了算法的局部搜索性能.

2.2 联赛选择方案(Tournament selection strategy)

定义 1 设$\varphi(f)$为当代群体适应度分布密度函数, $\overline{\varphi}'(f)$和$\overline{f}'$分别表示经过某种选择算子采样之后所生成下一代群体的期望适应度分布密度函数和期望适应度平均值, 则函数

$$V=\int_{-\infty}^{\infty}(f-\overline{f}')^2\overline{\varphi}'(f)\mathrm{d}f, \tag{6}$$

称为选择算子作用于适应度分布密度函数$\varphi(f)$的选择方差.

基变量联赛选择操作建立在个体适应度基础之上, 因此在应用某种选择算子对当前群体进行选择采样时, 其下一代群体中个体适应度的分布往往依赖与当代群体适应度分布. 为了能够对选择方法的选择方差进行定量的研究, 假定当前群体中个体适应度分布服从标准的正态分布$\overline{N}(0,1)$.

考虑当前群体适应度分布为离散变量的情形, 在N个个体中存在l个互异适应度函数值. 即$f_1<f_2<\cdots<f_l(n\leqslant N)$, 适应度分布函数为$\phi(f_i)$, $i=1,2,\cdots,l$.由联赛选择过程可知, 适应度等于或小于f_i的个体能够在联赛中胜出, 当且仅当所有参与联赛的个体的适应度必须小于或者等于f_i, 这也就意味在t次采样过程中, 每个个体的适应度必须等于或者是小于f_i. 下一代群体期望的适应度分布函数$\overline{\phi}'(f_i)$和适应度密度函数$\overline{\varphi}'(f_i)$可分别表示为

$$\overline{\phi}'(f_i)=\phi(f_i)^t,\overline{\varphi}'(f_i)=\phi(f_i)^t-\phi(f_{i-1})^t. \tag{7}$$

当适应度函数$\overline{\phi}'(f)$为连续变量时, 密度函数$\overline{\varphi}'(f)$可表示为

$$\overline{\varphi}'(f)=\frac{\mathrm{d}\overline{\phi}'(f)}{\mathrm{d}f}=t\varphi(f)\phi(f)^{t-1}. \tag{8}$$

进一步, 得到下一代适应度分布函数期望值$\overline{f}'$为

$$\overline{f}'=\int_{-\infty}^{\infty}tf\varphi(f)\phi(f)^{t-1}\mathrm{d}f. \tag{9}$$

当$t=2$时, 理论分析可得到$\overline{f}'=\sqrt{\dfrac{2}{\pi}}$.

根据定义1, 可以得出联赛规模为t的选择算子作用于$\varphi(f)$的选择方差V_t为

$$V_t=\int_{-\infty}^{\infty}t(f-\overline{f}')^2\varphi(f)\phi(f)^{t-1}\mathrm{d}f. \tag{10}$$

当联赛规模$t=1,2$, 且前一代群体适应度分布函数服从标准正态分布时, 计算得到

$$V_1=1,V_2=1-\frac{2}{\pi}. \tag{11}$$

同理, 根据式(10)可得到如下关系:

$$V_t<V_{t-1},t\geqslant 2. \tag{12}$$

从上式可以看出, 随着t的增大可取得越来越快的收敛速率, 但是对于多峰目标函数优化问题, 联赛选择算子较高的选择压力将会使算法收敛到某个局部最优解上, 这是造成算法早熟的一个重要因素.

3 自适应差分进化算法(Apaptive DE)

理论分析可知基变量的选取是影响算法性能的一个极其重要因素. 原DE算法中采用均匀选择机制选择基变量x_{r1}[5], 在牺牲一定局部搜索性能的前提下, 可有效的增加算法的全局搜索性能; Kaelo提出的DERL、DERB等算法[9]则采用联赛机制选取基变量x_{r1}, 即$x_{r1} = \min\{x_{i1}, x_{i2}, x_{i3} | i_1 \neq i_2 \neq i_3\}$, i表示当前正在处理的目标个体的下标, x_{i1}, x_{i2}, x_{i3}分别表示3个随机选取个体的下标. 这种基变量选取方式使得算法在进化后期可以加快算法的局部搜索能力. 但是对于一些高维复杂的多模优化问题, 由于其超曲面极其粗糙, 算法极可能在进化初期陷入某个局部最优解而丧失全局搜索能力. 自适应差分进化算法(ADE)采用带权质心$\overline{x}_i$ 作为对应于目标个体x_i变异操作中的基变量x_{r1}, 通过自适应调整权重(即调整基变量的选择压力)来实现算法从全局搜索与局部增强之间的平滑过渡.

3.1 加权质心基变量(Weighted centroid base vector selection)

假定当前目标个体为x_i, 3个随机选取的互异个体为x_{i1}, x_{i2}, x_{i3}, 则三点的质心$\widetilde{x}_i$ 可表示为

$$\widetilde{x}_i = \frac{1}{3}\sum_{j=1}^{3} x_{ij}. \tag{13}$$

考虑3个互异个体 x_{i1}, x_{i2}, x_{i3} 的目标函数值$f(x_{i1}), f(x_{i2}), f(x_{i3})$, 加权质心$\overline{x}_i$ 可定义为

$$\overline{x}_i = \sum_{j=1}^{3} \omega_j x_{ij}. \tag{14}$$

式中$\omega_j (j = 1, 2, 3)$可表示为

$$\omega_j = \frac{1}{\gamma_i (f(x_{ij}) - \widetilde{f}^i_{\min} + \sigma)}, \tag{15}$$

$$\gamma_i = \sum_{k=1}^{3} \frac{1}{f(x_{ik}) - \widetilde{f}^i_{\min} + \sigma}. \tag{16}$$

式中: $\widetilde{f}^i_{\min} = \min(f(x_{i1}), f(x_{i2}), f(x_{i3}))$, σ为收敛因子, 控制算法整体收敛速率, 定义为

$$\sigma = \alpha \left(\frac{1}{N-1} \sum_{i=1}^{N} (f_i - M)^2 \right)^{\beta}. \tag{17}$$

式中: N为群体规模, f_i表示第i个个体的目标函数, M 表示群体目标函数的平均值, α为一个足够大的正数, $\beta > 0$ 控制收敛因子变化的速率, α, β的选取应保证生成的初始群体满足如下条件

$$\sigma \gg f(x^0_{\max}) - f(x^0_{\min}). \tag{18}$$

$f(x^0_{\max})$和$f(x^0_{\min})$分别表示初始群体中目标值最大和最小的解. 这样在初始群体中由于$f(x^0_{\max}) > f(x_{ik}), f(x^0_{\min}) < \widetilde{f}^i_{\min}$, 可以得出$(k = 1, 2, 3)$

$$\sigma \gg f(x_{ik}) - \widetilde{f}^i_{\min}, i = 1, 2, \cdots, N. \tag{19}$$

在群体进化的初期, 公式(15)可转变为

$$\omega_j \approx \frac{1}{\sigma \times \frac{3}{\sigma}} = \frac{1}{3}, j = 1, 2, 3. \tag{20}$$

这将意味着基变量$x_{r1} = \overline{x}_i \approx \widetilde{x}_i$, 从而保证算法在初始阶段具有较好全局搜索能力(基变量随机选择策略). 随着算法的运行, 尤其是当群体中所有个体都聚集在某一个全局最优解附近区域时

$$\frac{1}{N-1} \sum_{i=1}^{N} (f_i - M)^2 \to 0. \tag{21}$$

收敛因子$\sigma \to 0$, 从而$x_{r1} = \overline{x}_i \to \widetilde{f}^i_{\min}$, 即在群体进化的后期, 算法逐渐倾向于选取3个互异个体中目标函数值最小的个体作为变异规则中的基变量(基变量联赛选择策略), 有效的增强算法的局部搜索能力.

进一步推广上述结果, 可以使原DE算法变异规则中的基变量x_{r1}为$m(3 < m < N-1)$个互异个体的带权质心$\overline{x}_i$, 即

$$x_{r1} = \overline{x}_i = \sum_{j=1}^{m} \omega_j x_{ij}, \tag{22}$$

其中ω_j根据公式(15)~(17)推广计算得到. 30个全局优化测试问题的数值计算结果表明$m = \max\{3, \mathrm{int}(0.1N)\}$ 是一个较好的选择.

3.2 加权变异算子(Weighted mutation operator)

假设每次从群体中选择$m(3 < m < N-1)$个个体, 采用m 个个体的加权质心$\overline{x}_i$ 作为基变量, 则原变异规则可转变为如下形式

$$\hat{x}_i = \overline{x}_i + F\vec{d}, \tag{23}$$

其中: $\vec{d}$为下降方向向量, F为增益算子.

为产生更好的点$\hat{x}_i$, 尽可能利用已经计算过的个体目标函数值. 假定当前从m 个子群体中选取了两个互异个体x_{ik1}和x_{ik2}, 如果$f(x_{ik1}) < f(x_{ik2})$, 则称$\vec{d} = x_{ik1} - x_{ik2}$ 为下降方向, 在此情况下, $\overline{x}_i$沿$\vec{d}$方向进行搜索将增加得到更好解的概率; 反之, 如果$f(x_{ik1}) > f(x_{ik2})$, 则下降方向$\vec{d} = x_{ik2} - x_{ik1}$.

3.3 反射和收缩算子(Reflection and contraction operator)

带权质心$\overline{x}_i$作为基变量对算法的收敛性能具有潜在地自适应调整作用, 该操作是在计算相应的测试点y_i 之前进行的. 在计算y_i之后, 如果$\widetilde{f}^i_{\min} = f(\tilde{x}^i_{\min}) < f(y_i) < f(x_i)$, 算法将利用$y_i$点对$\tilde{x}^i_{\min}$点执行反射和(或)收缩操作, 以期能够发现更好的解, 进一步增强算法局部收敛速度. 其中$\tilde{x}^i_{\min}$ 表示相对

于目标个体x_i所随机选取的m个个体中目标函数最小的个体. 反射操作和收缩操作分别定义如下:

$$r_i = \tilde{x}^i_{\min} - 0.5(y_i - \tilde{x}^i_{\min}), \tag{24}$$

$$c_i = \tilde{x}^i_{\min} + 0.5(y_i - \tilde{x}^i_{\min}). \tag{25}$$

通过反射和收缩操作, 就可以进一步搜索$\tilde{x}^i_{\min}$点附近区域. 由于$\tilde{x}^i_{\min}$为m个个体中最好的个体, 而不是整个群体N中最好的个体, 因此反射和收缩操作尽可能在不影响算法全局搜索能力前提下, 可以有效的增强算法局部搜索能力.

4　数值研究(Numerical experiments)

应用30个典型的全局优化问题(共32个实例)来保证数值研究结果客观性, 维数范围为2～10, 问题描述参见文献[10]. 每个问题具有不同的目标函数曲面地形和复杂度, 表1给出了30问题(P)的维数(n)及存在的极值点数(en), 其中“?”表示“未知”.

表 1　30个测试问题

Table 1　Thirty benchmark problems

P	n	en	P	n	en	P	n	en
ACK	10	?	AP	2	2	BL	2	4
BF1	2	?	BF2	2	?	BP	2	3
CB3	2	3	CB6	2	3	CM	2, 4	?
DA	2	3	EP	2	?	EM	5	?
EXP	10	?	GP	2	5	GW	10	?
GRP	3	?	H3	3	5	H6	6	5
HV	3	?	HSK	2	2	KL	4	?
LM1	3	125	LM2	5, 10	?	MC	2	2
MR	3	?	MCP	4	?	ML	10	?
MRP	2	3	MGP	2	5	NF2	4	?

ADE算法参数设置: 群体规模$N = 10n$; 选取规模$m = \max\{3, n\}$; 增益常数$F = 0.5$; 交叉概率$C_R = 0.5$; 收敛因子权重系数$\alpha = 1000$, 收敛因子速率控制系数$\beta = 3$. DE算法参数设置参见文献[5], DERL、DERB算法参数设置参见文献[9]. 此外, 4种算法的允许误差均设置为$\epsilon = 0.00001$, 评价指标采用目标函数评价数(function evaluation, fe)和成功率(success ratio, sr)来衡量, sr为成功运行次数与总运行次数之比.

对ADE、DE、DERL及DERB4种算法、32个实例(30个问题)各运行100次以消除运行过程中随机因素的影响, 即总共运行次数为$4 \times 32 \times 100 = 12800$. 为说明ADE算法对高维问题的有效性, 将30个问题分成A、B两组, A组包含了所有维数为2的实例, B组包含了所有维数大于2的问题实例. 表2、3给出了A、B组问题每种算法运行100次的平均结果.

表 2　A组问题计算结果

Table 2　The results of A group problems

P$-n$	DE		DERL		DERB		ADE	
	fe	sr	fe	sr	fe	sr	fe	sr
AP-2	789	1.00	709	1.00	660	1.00	562	1.00
BL-2	1193	1.00	783	1.00	857	1.00	655	1.00
BF1-2	1091	1.00	895	1.00	826	1.00	734	1.00
BF2-2	1767	1.00	986	1.00	977	1.00	792	1.00
BP-2	2561	1.00	896	1.00	934	1.00	682	1.00
CB3-2	1177	1.00	723	1.00	635	1.00	553	1.00
CB6-2	1595	1.00	829	1.00	740	1.00	590	1.00
CM-2	718	1.00	598	1.00	540	1.00	499	1.00
DA-2	1362	1.00	1005	1.00	904	1.00	890	1.00
EP-2	1235	0.93	807	1.00	704	1.00	629	1.00
GP-2	1949	1.00	919	1.00	795	1.00	692	1.00
HSK-2	689	1.00	614	1.00	538	1.00	474	1.00
MC-2	1141	1.00	663	1.00	602	1.00	508	1.00
MRP-2	4865	0.57	1359	0.60	1283	0.53	848	0.60
MGP-2	2421	0.83	827	0.47	770	0.53	558	0.60
AVE	1637	0.955	841	0.938	784	0.937	644	0.947

表 3　B组问题计算结果

Table 3　The results of B group problems

P$-n$	DE		DERL		DERB		ADE	
	fe	sr	fe	sr	fe	sr	fe	sr
ACK-10	41650	1.00	68030	1.00	41788	0.87	20855	1.00
EXP-10	14433	1.00	14286	1.00	5824	1.00	5040	1.00
GW-10	80283	1.00	98160	1.00	39831	0.47	17692	0.93
LM2-10	17610	1.00	17703	1.00	7727	1.00	5669	1.00
ML-10	100	0.00	100	0.00	100	0.00	100	0.00
H6-6	10110*	0.70*	6820	0.90	3756	0.60	2922	0.93
EM-5	21843	1.00	8820	0.87	6041	0.63	5224	0.80
LM2-5	4827	1.00	4070	1.00	2671	1.00	2211	1.00
CM-4	2972	1.00	2504	1.00	1709	1.00	1736	1.00
KL-4	89524	1.00	5653	1.00	3461	1.00	2060	1.00
MCP-4	54947	1.00	2990	1.00	2329	1.00	2247	1.00
NF2-4	780745	1.00	67874	0.10	49405	0.97	14550	1.00
GRP-3	42995	1.00	5329	1.00	5052	1.00	1783	1.00
H3-3	2062	1.00	1364	1.00	1090	1.00	789	1.00
HV-3	2254	1.00	1834	1.00	1452	1.00	1319	1.00
LM1-3	1844	1.00	1541	1.00	1227	1.00	1137	1.00
MR-3	20854	1.00	3623	1.00	3273	1.00	2296	1.00
AVE	69944	0.923	18277	0.875	10396	0.855	5155	0.921

*: $\epsilon = 0.20$

表2说明A组15个二维问题, ADE算法平均函数评价数为644, 分别相当于DE、DERL及DERB算法平均函数评价数的39.3%, 76.6%和82.1%; 而且ADE算法的可靠性也高于DERL、DERB算法, 稍

低于DE算法. 可以看出ADE算法在成功率较高的前提下, 计算问题得到较大的改善.

表3说明B组17个(维数为3～10)问题求解实例. ADE算法平均函数评价数为5155, 分别相当于DE、DERL及DERB算法平均函数评价数的7.4%, 28.2%, 和49.6%, 可靠性基本与DE算法相同, 且高于DERL、DERB算法. 此外, ADE算法B组的计算结果相对于A组而, 其计算代价改善的幅度更大, 性能得到了进一步的改善. 由此可以得知ADE算法更适合于高维复杂问题的求解. 需要特别指出的是ML-10和H6-6两个问题, ML-10问题目标函数曲面为脉冲山峰, 由于采用有限的群体规模($N = 100$), 群体差异基本上为0, 4种算法均无法得到问题的全局最优解; 另外, DE算法在求解H6-6问题时发现, 当允许误差$\epsilon = 0.00001$时, 算法基本上无法终止, 因此DE算法求解该问题时, 将允许误差设置为$\epsilon = 0.20$.

5 结论(Conclusion)

不损害全局搜索能力的前提下, 如何提高算法局部增强能力是多模全局优化领域中的亟需解决的一个重要问题. 在基变量选择策略理论分析基础上, 本文提出一种自适应差分进化算法ADE, 通过构建自调整的权重变异策略, 实现算法从全局搜索到局部搜索之间的平滑过渡, 另外基于下降方向的启发式规则及单纯形操作算子也进一步增强了在极值点附近的局部搜索能力. 数值研究结果表明ADE算法有效性.

参考文献(References):

[1] MORDECAL A. *Nonlinear Programming Analysis and Methods*[M]. New York, American: Prentice Hall, 1976.

[2] NELDER J A, MEAD R A. A simplex method for function minimization[J]. *Computer Journal*, 1965, 7(7): 308 – 313.

[3] WALSH G R. *Methods of Optimization*[M]. London, England: Wiley Press, 1975.

[4] HORST R, TUY H. Global Optimization[M]// *Deterministic Approaches*. Berlin, Germany: Springer Verlag, 1990.

[5] STORN R. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces[J]. *Journal of Global Optimization*, 1997, 11(4): 341 – 359.

[6] ALI M M, TÖRN A. Population set based global optimization algorithms: some modification and numerical studies[J]. *Computer and Operations Research*, 2004, 31(10): 1703 – 1725.

[7] GOLDBERG D E. *Genetic Algorithm in Search, Optimization and Machine Learning*[M]. Boston, American: Addison-Wesley, 1989.

[8] ALI M M, TÖRN A. A numerical comparison of some modified controlled random search algorithms[J]. *Journal of Global Optimization*, 1997, 11(4): 377 – 385.

[9] KAELO P, ALI M M. A numerical study of some modified differential evolution algorithms[J]. *European Journal of Operational Research*, 2006, 169(3): 1176 – 1184.

[10] ALI M M, KHOMPATRAPORN C, ZABINSKY Z B. A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems[J]. *Journal of Global Optimization*, 2005, 31(4): 635 – 672.

[11] ROGERS A, PRUGER-BENNETT A. Genetic drift in genetic algorithm selection schemes[J]. *IEEE Transactions on Evolutionary Computation*, 1999, 3(4): 298 – 303.

作者简介:

张贵军 (1974—), 男, 浙江工业大学副教授, 主要研究方向为优化调度等, E-mail: zgj@zjut.edu.cn;

王信波 (1984—), 男, 浙江工业大学硕士研究生, 主要要就方向优化调度等, E-mail: roomct@hotmail.com;

俞 立 (1961—), 男, 浙江工业大学教授, 博士生导师, 主要研究方向网络控制系统等, E-mail: lyu@zjut.edu.cn;

冯远静 (1976—), 男, 浙江工业大学副教授, 主要研究方向随机优化等, E-mail: fyjing@zjut.edu.cn.