

多变异策略的自适应差分演化算法

周雅兰¹ 徐志²

(广东财经大学信息学院 广州 510320)¹ (中山大学信息科学与技术学院 广州 510006)²

摘要 差分演化(Differential Evolution, DE)算法的性能依赖于变异策略的选择和控制参数的设置。不同问题对 DE 的变异策略和参数的设置各不相同。为了提高 DE 的性能,提出一种多变异策略的自适应差分演化算法,建立由多种变异策略组成的策略池,两个主要参数自适应策略控制。为了验证所提算法的性能,在测试数据集 CEC2013 上进行了实验,并将其与使用 6 种不同变异策略的原始 DE 和 4 种改进 DE 进行比较。实验结果表明,提出的算法是一种有效的 DE 变种,其性能优于其它 DE。

关键词 差分演化算法,多变异策略,参数自适应

中图分类号 TP183 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.6.052

Self-adaptive Differential Evolution with Multi-mutation Strategies

ZHOU Ya-lan¹ XU Zhi²

(School of Information Science, Guangdong University of Finance & Economics, Guangzhou 510320, China)¹

(School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510006, China)²

Abstract The performance of differential evolution(DE) algorithm often depends heavily on the mutation strategy and control parameters. A novel self-adaptive differential evolution with multi-mutation strategies called SMSDE was proposed. SMSDE designs a strategy pool consisting of many kinds of mutation strategy and applies self-adaptive strategies to two main parameters. In order to verify the performance of SMSDE, SMSDE was compared with 6 original DEs and 4 advanced DEs on CEC2013 benchmark functions. The experimental results show that SMSDE is superior to original DEs, and is competitive with the current advanced DE variants.

Keywords Differential evolution algorithm, Multi-mutation strategies, Parameter self-adaptation

1 前言

差分演化^[1]算法是由 Storn 和 Price 于 1997 年提出的一种基于种群的全局演化算法,因其简单的操作和有效的优化效果受到研究者的广泛关注,并被大量地用于解决科学和工程领域的优化问题。原始 DE 主要有 3 个演化操作:变异、交叉和选择,通过这 3 个演化操作促使种群向全局最优移动,其中变异操作以及变异和交叉的控制参数对 DE 算法的性能影响较大。近年,对 DE 算法的改进可分成 4 类:1)新候选解的产生操作(包括变异与交叉操作)的改进。文献[2]提出了基于距离的变异操作。文献[3]引入多目标的方法,选择适应度值和保持多样性较好的个体用于变异。文献[4]不仅利用邻域个体的信息在最优个体附近进行开采,加速算法收敛,而且引入方向信息来阻止搜索进入差解区域并移动到有潜在解的区域进行探索。文献[5]提出了自适应策略的 DE 算法。文献[6]提出了改进的交叉操作。2)控制参数的设置。这方面成果很多,如文献[7]提出了自适应参数控制方法 jDE。文献[8]综述了所有参数控制方法。3)新候选解的产生操作和参数控制集成。Qin 和 Suganthan 提出变异向量的产生操作

和控制参数集成的 SaDE^[9]。王勇等人提出 CoDE,将 3 种常用变异策略与 3 组控制参数随机配对生成试验向量^[10]。Zhang 等人提出带自适应参数和改进变异策略的 JADE^[11]。龚文引等人采用 4 种变异策略的产生操作并与文献[11]相似的参数自适应方法相集成^[12]。文献[13]在交叉操作中引入协方差矩阵学习,并与双模分布控制参数的方法相集成。4)混合其它方法。混合其他智能优化方法可以进一步提升 DE 性能,如文献[14]综述了 DE 和粒子群算法的各种混合版本。其中, jDE、SaDE、CoDE 和 JADE 与本文提出的算法关联较大,将在 2.2 节中进一步介绍。除了算法改进,文献[15]对 DE 的收敛性作了初步的研究。DE 广泛应用于实际的优化问题,如文献[16,17]。Das 和 Suganthan 对 2011 年以前 DE 的理论、改进和应用情况进行了综述^[18]。

本文提出一种新的 DE 改进算法,称为多变异策略的自适应差分演化算法(Self-adaptive DE with Multi-mutation Strategies, SMSDE)。提出的 SMSDE 建立一个策略池,选用 3 种具有不同偏好的改进变异策略放入池中,变异缩放因子 F 用基于柯西分布的自适应策略控制,交叉概率 CR 用一种高斯分布的变形实现自适应控制。此外,为了维护种群的多

到稿日期:2014-07-02 返修日期:2014-10-13 本文受广州市珠江科技新星专项(2012J2200085),广东省教育厅高校优秀青年创新人才培养项目(2012LYM_0066),广东商学院科研创新团队建设计划资助。

周雅兰(1979-),女,博士,副教授,CCF 会员,主要研究方向为人工智能与数据挖掘,E-mail:zhouyulan@163.com;徐志(1989-),男,硕士生,主要研究方向为人工智能。

样性, SMSDE 将种群的均值点作为种群中的一个个体。在 CEC2013 测试函数集上进行了实验, 并分别与使用 6 种不同变异策略的原始 DE 和 4 种具有代表性的改进 DE 进行了比较, 实验结果表明了所提算法的有效性。

2 差分演化算法

2.1 原始差分演化算法

原始 DE 主要有 3 个演化操作: 变异、交叉和选择^[1]。

变异: 假设在 D 维空间, 第 g 代的解向量(也称目标向量)为 $X_i^g = (x_{i1}^g, \dots, x_{id}^g, \dots, x_{iD}^g)$, $i=1, 2, \dots, NP$; 变异向量为 $V_i^g = (v_{i1}^g, \dots, v_{id}^g, \dots, v_{iD}^g)$, 常用的变异策略有 6 种^[2], 用“DE/a/b”来表示, 其中 DE 表示差分演化算法, a 表示基向量的选择方式, b 表示算子中差分向量的个数: 1) DE/rand/1: $V_i^g = X_{r_1}^g + F \cdot (X_{r_2}^g - X_{r_3}^g)$; 2) DE/rand/2: $V_i^g = X_{r_1}^g + F \cdot (X_{r_2}^g - X_{r_3}^g) + F \cdot (X_{r_4}^g - X_{r_5}^g)$; 3) DE/best/1: $V_i^g = X_{best}^g + F \cdot (X_{r_1}^g - X_{r_2}^g)$; 4) DE/best/2: $V_i^g = X_{best}^g + F \cdot (X_{r_1}^g - X_{r_2}^g) + F \cdot (X_{r_3}^g - X_{r_4}^g)$; 5) DE/current-to-best/1: $V_i^g = X_i^g + F \cdot (X_{best}^g - X_i^g) + F \cdot (X_{r_1}^g - X_{r_2}^g)$; 6) DE/rand-to-best/1: $V_i^g = X_{r_0}^g + F \cdot (X_{best}^g - X_{r_0}^g) + F \cdot (X_{r_1}^g - X_{r_2}^g)$, 其中 X_{r_0} 是完全随机选择的父辈个体, $r_0 \neq r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5 \neq i \in \{1, 2, \dots, NP\}$; X_{best} 是最优个体; 参数 $F \in (0, 1]$ 为缩放因子。上述变异算子中的变异向量都是由缩放后的差分向量加载在一个基向量上而成的。

交叉: 将目标向量 X_i^g 和变异向量中的维值交叉, 用于产生实验向量 $Trial_i^g = (trial_{i1}^g, \dots, trial_{id}^g, \dots, trial_{iD}^g)$ 。交叉方式有指数交叉法和二项式交叉法两种, 常用的是二项式交叉法, 描述如下:

$$trial_{id}^g = \begin{cases} v_{id}^g, & \text{if } \text{rand}() < CR \text{ or } d = k \\ x_{id}^g & \text{otherwise} \end{cases} \quad (1)$$

其中, $CR \in [0, 1]$ 是交叉概率, $k \in \{1, 2, \dots, D\}$ 是随机选择的维。

选择: 采用一对一贪婪选择算子如下:

$$X_i^{g+1} = \begin{cases} Trial_i^g, & \text{if } Trial_i^g \text{ is better than } X_i^g \\ X_i^g, & \text{otherwise} \end{cases} \quad (2)$$

2.2 改进的差分演化算法

差分演化算法对于变异策略的选择和控制参数的设置敏感, 针对这两个方面有许多改进算法。下面对这两方面分别选择性能较好的几种改进算法进行简述, 分别是 SaDE、CoDE 与 jDE、JADE。

SaDE^[9] (Self-Adaptive DE) 是首次提出的一种多变异策略的差分演化算法。SaDE 建立一个变异策略的候选池, 将“DE/rand/1”、“DE/rand-to-best/1”、“DE/rand/2”、“DE/current-to-rand/1”4 种变异策略加入到候选池中, 各变异策略有各自的被选择概率, 这些概率在演化过程中根据以往的成功经验进行动态调整。

CoDE^[10] (Composite DE) 是王勇等人提出的一种组合多变异策略的差分演化算法。CoDE 将 3 种变异策略与 3 组控制参数随机配对生成实验个体, 即每个目标个体在生成后代个体阶段需要经历 3 次变异操作, 产生 3 个实验向量, 再从这 3 个实验向量中选择最优的一个进入选择阶段。CoDE 采用的 3 种变异操作为“DE/rand/1”、“DE/rand/2”、“DE/current-to-rand/1”, 3 组控制参数分别为: $F=1.0, CR=0.1; F=1.0,$

$CR=0.9; F=0.8, CR=0.2$ 。

jDE^[7] (Janzest Brest DE) 是 Brest 提出的一种对每个个体的 F 值和 CR 值进行自适应的算法。jDE 每一轮演化开始时每个个体的 F 和 CR 分别以一定概率产生符合均匀分布的随机扰动, 当个体在这一轮演化中成功被更新时, 扰动产生的 CR 和 F 值被保留下来进入下一轮演化, 否则恢复到之前的 F 值和 CR 值。

JADE^[11] (Jingqiao Zhang Adaptive DE) 是 Zhang 和 Sanderson 提出的一种自适应差分演化算法, 其 F 和 CR 分别从柯西分布 $C(\mu_F, 0, 1)$ 和高斯分布 $N(\mu_{CR}, 0, 1)$ 中随机产生, 并编码到每一个个体中, 其中参数 μ_F 和 μ_{CR} 的初始值均为 0.5。JADE 算法在每一次演化开始时利用存储在集合 goodF 和 goodCR 中的 F 值和 CR 值更新参数 μ_F 和 μ_{CR} , goodF 和 goodCR 中存储的是上一代中成功被更新的个体的 F 值和 CR 值。

3 提出的 SMSDE

原始 DE 算法的不同变异策略具有不同的偏好^[2]。2.1 节所述的 6 种常用变异策略中, “DE/best/1”的局部搜索能力最强, 搜索范围最窄, 有利于种群收敛, 但容易导致过早收敛或者陷入局部最优; “DE/rand/2”的全局搜索能力最强, 搜索范围最广, 几乎覆盖了所有决策空间, 而且搜索的点分布也比较均匀, 但收敛速度很慢, 不利于种群的快速收敛; 另外 4 种变异策略的搜索能力介于“DE/best/1”和“DE/rand/2”之间。DE 算法的性能对缩放因子 F 和交叉概率 CR 的设置非常敏感, 而自适应策略是实现参数控制最有效的方法之一^[8]。为了使算法对于不同的问题都有较好的效果, 提高 DE 算法的性能和通用性, 本文提出了采用多种变异策略和参数自适应控制策略的 SMSDE。

3.1 多变异策略

SMSDE 建立了一个策略池, 策略池由多种变异策略组成, 演化时每个个体从策略池中随机选择一种变异策略进行变异操作。SMSDE 选择如下 3 种变异策略放在策略池中: “DE/rand-to-pbest/1”、“DE/current-to-pbest/1”、“DE/current-to-rand/1”。其中“DE/rand-to-pbest/1”和“DE/current-to-pbest/1”分别为“DE/rand-to-best/1”和“DE/current-to-best/1”的变种^[12], 它们的 $pbest$ 不是单一的 $best$ 个体, 而是从前 $p100\%$ 的最佳个体中随机选取的个体, p 在 $(0, 1]$ 之间取值。这样“DE/rand-to-pbest/1”利用了多个优秀个体信息, 使得算法不至于过早收敛, 且由于“DE/rand-to-pbest/1”的基向量为随机选取, 因此较其它利用 $best$ 个体信息的变异策略(如“DE/best/1”等)而言其全局搜索能力更强。“DE/current-to-pbest/1”比“DE/rand-to-pbest/1”的局部搜索能力更强, 且擅长解决多峰和不连续问题。“DE/current-to-rand/1”为所选的 3 个变异策略中全局搜索能力最强的变异策略^[19]。当 $CR=1$ 时, “DE/current-to-rand/1”进行的是算术交叉, 具有旋转不变的特性, 对于解决旋转问题有优势。

“DE/rand-to-pbest/1”、“DE/current-to-pbest/1”和“DE/current-to-rand/1”三者以随机的方式混合时, 搜索的范围广泛但不至于过于随机, 搜索的粒度在目标个体和种群最优个体附近较细, 有利于种群的收敛, 能够较好地平衡全局搜索和局部搜索能力。

3.2 参数 F 和 CR 的自适应控制

缩放因子 F 的生成方式采用 JADE 中的方法通过柯西

分布随机产生^[11]。选用这种自适应方式一方面考虑到变异策略中使用了“DE/current-to-pbest/1”，而前人的经验表明基于柯西分布的自适应对于变异策略“DE/current-to-pbest/1”有较好的效果^[12]；另一方面，柯西分布相对于其他分布而言能够获得更广范围的 F 随机值，但又不会像完全随机一样没有侧重点，因此有更多的机会找到最佳 F 值。具体公式如下^[11]：

$$F_i = C_i(F_m, 0, 1) \quad (3)$$

$$F_m = (1-c) \times F_m + c \times \text{mean}_L(F_{suc}) \quad (4)$$

其中，参数 c 为学习因子在 $(0, 1)$ 之间取值， F_m 初始值为 0.5 ^[11]。 F_{suc} 代表成功的 F_i 值集合，通过这些 F_i 值产生的实验个体能够成功取代父代个体进入下一代。 $\text{mean}_L(\cdot)$ 代表

$$\text{Lehmer 平均, 公式为 } \text{mean}_L(F_{suc}) = \frac{\sum_{F \in F_{suc}} F^2}{\sum_{F \in F_{suc}} F}。$$

前人的经验表明^[19,20]，交叉概率 CR 值处于 $[0, 1]$ 的两端较为有效，且 CR 值较小时有利于求解可分问题，值较大时有利于解决不可分问题。因此 CR 的取值应该处于 0 或者 1 的附近。基于此，提出一种新的分布模型用于 CR 的自适应，这种分布模型的公式为：

$$CR_i = \begin{cases} CR_{t_i}, & \text{if } CR_{t_i} \geq 0 \\ 1 + CR_{t_i}, & \text{if } CR_{t_i} < 0 \end{cases} \quad (5)$$

$$CR_{t_i} = N_i(0, CR_d) \quad (6)$$

标准差 CR_d 的更新公式如下：

$$CR_d = (1-c) \times CR_d + c \times \text{std}(\widetilde{CR}_{suc}) \quad (7)$$

其中，参数 c 为学习因子，在 $(0, 1)$ 之间取值， CR_d 初始值与式(4)中的 F_m 一样，取 0 和 1 之间的中间值 0.5 。 \widetilde{CR}_{suc} 表示成功的 CR_i 值集合，通过这些 CR_i 值产生的实验个体能够成功取代父代个体进入下一代。 \widetilde{CR}_{suc} 为 CR_{suc} 中的个体 CR_i 经过如下公式处理：

$$CR_i = \begin{cases} CR_i, & \text{if } CR_i \leq 0.5 \\ 1 - CR_i, & \text{if } CR_i > 0.5 \end{cases} \quad (8)$$

通过式(5)和式(6)生成的 CR 分布在 $[0, 1]$ 两端，如图 1 所示。

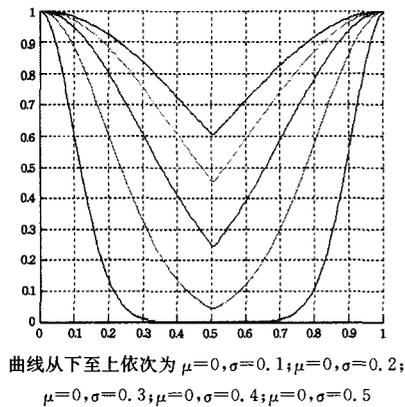


图 1 CR 取值的分布

3.3 SMSDE 算法

图 2 为 SMSDE 算法流程图。SMSDE 从策略池中均匀随机选择变异策略，保证了每个变异策略被选择的概率是相同的，使变异策略池中的各个变异策略的偏好都能得到充分利用。采用柯西分布和高斯分布的一种变形分别实现了 F 和 CR 参数值的自适应控制。此外，种群的均值点也作为种

群的一个个体，由于均值点不是通过变异和交叉操作产生的，且更新方式是求新的种群均值点，不受变异策略以及控制参数 F 和 CR 值的影响，因此加入该个体能够维护种群的多样性。

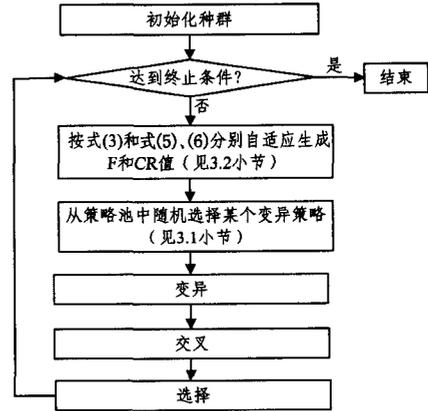


图 2 SMSDE 算法流程

相对于前人提出的自适应算法^[7,9-11]，SMSDE 算法采用了更为先进的多种变异策略，并采用简单的随机混合策略；在参数自适应方面，SMSDE 一方面利用了前人的方法设置 F 值，另一方面利用了前人的经验提出了一种全新的自适应 CR 的方法，使得 CR 处于 0 或 1 附近，能够有效地解决各种类型的问题。此外，SMSDE 算法的两个参数 c 和 p 与具体问题无关，算法的复杂度并没有相应增加，结构简单，易于实现。

4 实验结果及分析

为了验证本文所提算法的有效性，在 CEC2013^[21] 的 28 个测试函数上进行了仿真实验，其中 $F1-F5$ 为单峰函数， $F6-F20$ 为基本多峰函数， $F21-F28$ 为复杂函数，所有函数的个体维数设定为 $D=30$ 。算法测试采用 Matlab 在 PC (CPU Intel E-2630, 2.3GHz) 上运行。

4.1 算法参数的设置

提出的 SMSDE 算法实现了参数 F 和 CR 的自适应，用与具体问题无关的两个新参数 c 和 p 分别控制参数的自适应率和变异策略的贪婪程度。 c 为参数 F 和 CR 的学习因子，用于平衡参数学习过程中的新旧知识，当 $c=0$ 时算法忽略新的知识，从而没有学习新知识的能力；当 $c=1$ 时算法遗忘旧知识， F 和 CR 不能平滑地更新，从而导致参数震荡。 p 决定利用多少优秀个体的信息用于变异策略，其值越小，变异策略越贪婪，降低了种群的多样性，如 $p=0.01$ ， $NP=100$ 时， $p \times NP=1$ ，只有单一的优秀个体引导变异，使得种群过早收敛。实验表明，通常 c 在 $[0.05, 0.2]$ 、 p 在 $[0.05, 0.2]$ 之间取值时，SMSDE 算法的性能较好。这与其它文献如文献^[11]中的实验结论一致，限于篇幅，本文没有列出具体的实验结果，具体可参见文献^[11]的参数讨论。因为 SMSDE 中参数 c 和 p 的设置与 JADE 类似，所以在实验中，SMSDE 的 $p=0.05$ ， $c=0.1$ ， NP 的值与文献^[7,11]一样设为 100 。

6 种不同变异策略的原始 DE 的参数设置为 $F=0.5$ ， $CR=0.9$ ， $NP=100$ ；4 种改进 DE 的参数设置同原文^[7,9-11]；所有算法的停止条件是目标函数评估次数为 $10^4 \times D$ ，其中 D 为基准测试函数个体的维度，即目标函数评估 $10^4 \times 30$ 次后算法停止；每个算法独立运行 51 次后对所有数据进行统计分析，在每一个问题上进行了 Wilcoxon signed ranks ($p=5\%$) 测试^[22]。

4.2 与6种不同变异策略的原始DE比较

表1是SMSDE与6种不同变异策略的原始DE的比较结果。限于篇幅,表1中省略了具体的函数值,仅列出了统计的值,其中“1”、“0”、“-1”分别表示SMSDE在这个函数上的表现相对比算法在统计意义上更优秀、相似和更差;符号w/t/l分别表示SMSDE的表现相对比算法更优秀、相似和更差的函数个数。

表1 与6种不同变异策略的原始DE的比较结果

函数	DE/ rand/1	DE/ rand/2	DE/ best/1	DE/ best/2	DE/current- to-best/1	DE/rand- to-best/1
F1	0	1	1	1	1	1
F2	1	1	1	1	1	1
F3	-1	1	1	1	1	1
F4	1	1	-1	1	1	1
F5	1	1	1	1	1	1
F6	1	1	1	-1	1	1
F7	-1	1	1	1	1	1
F8	1	1	1	1	1	1
F9	1	1	1	-1	1	1
F10	-1	1	1	-1	1	1
F11	1	1	1	1	1	1
F12	1	1	1	1	1	1
F13	1	1	1	1	1	1
F14	1	1	1	1	1	1
F15	1	1	1	1	1	1
F16	1	1	1	1	1	1
F17	1	1	1	1	1	1
F18	1	1	1	1	1	1
F19	1	1	1	1	1	1
F20	1	1	1	1	1	1
F21	0	1	1	0	1	1
F22	1	1	1	1	1	1
F23	1	1	1	1	1	0
F24	-1	1	1	1	1	1
F25	-1	1	1	1	1	1
F26	1	-1	1	1	1	1
F27	-1	1	1	1	1	1
F28	0	1	1	1	1	1
w/t/l	19/3/6	27/0/1	27/0/1	24/1/3	28/0/0	27/1/0

从表1可以看出,SMSDE在19个测试函数上优于“DE/rand/1”,6个测试函数上明显不如“DE/rand/1”,3个测试函数上与“DE/rand/1”表现相似;在27个测试函数上的表现比“DE/rand/2”、“DE/best/1”和“DE/rand-to-best/1”优秀,分别在函数F26和F4上表现不如“DE/rand/2”和“DE/best/1”,在函数F23上表现与“DE/rand-to-best/1”相似;在24个测试函数上表现优于“DE/best/2”,在F6、F9和F10这3个函数上不如“DE/best/2”,在函数F21上表现与之相似;在全部的28个测试函数上表现得比“DE/current-to-best/1”优秀。

由此可见,SMSDE在解决单峰问题和基本多峰问题上的表现比“DE/rand/1”优秀,而在解决复杂问题上与“DE/rand/1”各有优劣;在测试的大部分问题上的表现比“DE/rand/2”优秀;在测试函数上整体的表现比“DE/best/1”和“DE/best/2”优秀,可知SMSDE的局部搜索能力不比“DE/best/1”和“DE/best/2”弱,而全局搜索能力则比“DE/best/1”和“DE/best/2”强;SMSDE由于在全局和局部搜索方面都有不错的表现,因此比较容易表现得比“DE/current-to-best/1/bin”和“DE/rand-to-best/1/bin”优秀。

通过以上对表1的对比结果分析可知,SMSDE在单峰问题、基本多峰问题以及复杂问题上表现优于原始DE,原因是其较好地平衡了全局搜索能力和局部搜索能力。

4.3 与4种改进DE的比较

表2为SMSDE与4种改进DE算法,SaDE、CoDE、jDE和JADE的比较结果。表2中省略了具体函数值。SaDE为自适应参数和多变异策略的差分演化算法,算法思想与提出的SMSDE比较相似。CoDE为多变异策略的差分演化算法,其算法思想更倾向于利用前人的知识提升优化效果,而提出的SMSDE算法则更倾向于自身演化过程中的信息利用。jDE和JADE是只用了一种变异策略的自适应差分演化算法。与SaDE相比,SMSDE在23个测试函数上表现得比SaDE优秀,在2个测试函数上表现得相似,在2个测试函数上不如SaDE。与CoDE相比,SMSDE在17个测试函数上表现得比CoDE优秀,在5个测试函数上表现得相似,在6个测试函数上不如CoDE。与jDE相比,SMSDE在19个测试函数上表现得比jDE优秀,在6个函数上表现得相似,在3个函数上表现不如jDE。与JADE相比,SMSDE在15个测试函数上表现得比JADE优秀,在4个测试函数上表现得相似,在9个测试函数上表现不如JADE。

表2 SMSDE与4种改进DE算法的比较结果

函数	SaDE	CoDE	jDE	JADE
F1	0	0	0	0
F2	1	1	1	-1
F3	1	1	1	1
F4	1	-1	1	1
F5	-1	-1	1	1
F6	1	-1	1	-1
F7	1	1	1	1
F8	1	0	1	1
F9	1	1	1	1
F10	1	0	0	0
F11	1	0	0	0
F12	1	1	1	-1
F13	1	1	1	-1
F14	1	1	-1	-1
F15	1	1	1	1
F16	1	1	1	1
F17	0	1	-1	-1
F18	1	1	1	1
F19	1	-1	-1	-1
F20	1	1	1	1
F21	1	-1	0	-1
F22	1	1	1	-1
F23	1	1	1	1
F24	1	1	1	1
F25	1	1	0	1
F26	-1	-1	1	1
F27	1	1	1	1
F28	0	0	0	0
w/t/l	23/3/2	17/5/6	19/6/3	15/4/9

从表2可知,整体而言SMSDE比SaDE和CoDE优秀,表明与多变异策略的改进DE相比,SMSDE具有很强的竞争力。SMSDE拥有不弱于jDE和JADE的优化能力,表明SMSDE相比参数自适应的改进DE具有很强的竞争力。

结束语 差分演化算法是一种简单而有效的演化算法,其应用相当广泛。但差分演化算法的性能受变异策略选择和参数设置的影响,为了提高算法的性能和通用性,本文设计由3种不同偏好变异策略组成的策略池并采用参数自适应控制策略,提出了多变异策略的自适应差分演化算法,简称SMS-DE。SMSDE在CEC2013测试函数集上进行了实验,并与采

(下转第255页)

- [5] 徐章艳,杨炳儒,宋威. 基于简化的二进制差别矩阵的快速属性约简算法[J]. 计算机科学,2006,33(4):65-68
Xu Zhang-yan, Yang Bing-ru, Song Wei. Fast attribute reduction algorithm based on simple binary discernibility matrix[J]. Computer science, 2006, 33(4): 65-68
- [6] 葛浩,李龙澍,杨传健. 改进的快速属性约简算法[J]. 小型微型计算机系统,2009,30(2):308-312
Ge Hao, Li Long-shu, Yang Chuan-jian. Advanced Fast attribute reduction algorithm[J]. Micro computer system, 2009, 30(2): 308-312
- [7] 刘宗田. 属性最小约简的增量式算法[J]. 电子学报,1999,27

(11):96-98

Liu Zong-tian. An incremental algorithm for minimum attribute reduction[J]. Chinese Journal of Electronics, 1999, 27(11): 96-98

- [8] Jelonek J, Krawiec K, Slowinski R. Rough set reduction of attributes and their domains for neural networks[J]. Computational Intelligence, 1995, 11(2): 339-347
- [9] 杨明. 一种基于改进的差别矩阵的属性约简增量式更新算法[J]. 计算机学报,2007,30(5):815-822
Yang Ming. An incremental updating algorithm for attribute reduction based on improved discernibility matrix [J]. Chinese Journal of computers, 2007, 30(5): 815-822

(上接第 250 页)

用 6 种不同变异策略的原始 DE 以及 4 种有代表性的改进 DE 进行了比较。实验结果表明, SMSDE 的优化效果优于其他对比的算法。

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] Eptropakis M G, Tasoulis D K, Pavlidis N G, et al. Enhancing differential evolution utilizing proximity-based mutation operators [J]. IEEE Transactions on Evolutionary Computation, 2011, 15(1): 99-119
- [3] Wang J, Liao J, Zhou Y, et al. Differential evolution enhanced with multiobjective sorting based mutation operators[J]. IEEE Trans. Cybernetics, 2014, 46(12): 2792-2805
- [4] Cai Y, Wang J. Differential evolution with neighborhood and direction information for numerical optimization[J]. IEEE Trans. Cybernetics, 2013, 43(2): 634-647
- [5] 薛羽, 庄毅, 顾晶晶, 等. 自适应离散差分进化算法策略的选择[J]. 软件学报, 2014, 25(5): 984-996
Xue Yu, Zhuang Yi, Gu Jing-jing, et al. Strategy selecting problem of self-adaptive discrete differential evolution algorithm[J]. Journal of Software, 2014, 25(5): 984-996
- [6] Guo S, Yang C. Enhancing differential evolution utilizing eigenvector-based crossover operator[J]. IEEE Transactions on Evolutionary Computation, 2014, 19(1): 31-49
- [7] Brest J, Greiner S, Boskovic B, et al. 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] 杨振宇, 唐柯. 差分进化算法参数控制与适应策略综述[J]. 智能系统学报, 2011, 6(5): 415-423
Yang Zhen-yu, Tang Ke. An overview of parameter control and adaptation strategies in differential evolution algorithm [J]. CAAI Transactions on Intelligent Systems, 2011, 6(5): 415-423
- [9] Qin A K, Huang V L, Suganthan P N. Differential evolution algorithm with strategy adaptation for global numerical optimization [J]. IEEE Transactions on Evolutionary Computation, 2009, 13(2): 398-417
- [10] Wang Y, Cai Z, Zhang Q. Differential evolution with composite trial vector generation strategies and control parameters [J]. IEEE Transactions on Evolutionary Computation, 2011, 15(1): 55-66
- [11] Zhang J, Sanderson A C. JADE: adaptive differential evolution

with optional external archive[J]. IEEE Transactions on Evolutionary Computation, 2009, 13(5): 945-958

- [12] Gong W, Cai Z, Ling C X, et al. Enhanced differential evolution with adaptive strategies for numerical optimization [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2011, 41(2): 397-413
- [13] Wang Y, Li H-X, Huang T, et al. Differential evolution based on covariance matrix learning and bimodal distribution parameter setting[J]. Applied Soft Computing, 2014, 18: 232-247
- [14] Xin B, Chen J, Zhang J, et al. Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: a review and taxonomy [J]. IEEE Trans Systems Man Cybern Part C Appl Rev, 2011, 42(5): 744-767
- [15] Ghosh S, Das S, Vasilakos A V, et al. On convergence of differential evolution over a class of continuous functions with unique global optimum [J]. IEEE Trans. Syst., Man, Cybern. B, Cybern., 2012, 42(1): 107-124
- [16] Ding Y, Jiao Y-C, Zhang L, et al. Solving port selection problem in multiple beam antenna satellite communication system by using differential evolution algorithm[J]. IEEE Transactions on Antennas and Propagation, 2014, 62(10): 5357-5361
- [17] 毕志升, 王甲海, 印鉴. 基于差分演化算法的软子空间聚类[J]. 计算机学报, 2012, 35(10): 2116-2128
Bi Zhi-sheng, Wang Jia-hai, Yin Jian. Subspace clustering based on differential evolution[J]. Chinese Journal of Computer, 2012, 35(10): 2116-2128
- [18] Das S, Suganthan P N. Differential evolution: A survey of the state-of-the-art[J]. IEEE Transactions on Evolutionary Computation, 2011, 15(1): 4-31
- [19] Mezura-Montes E, Velázquez-Reyes J, Coello Coello C A. A comparative study of differential evolution variants for global optimization [C]//Proceedings of 8th Annual Conference on Genetic and Evolutionary Computation. ACM, 2006: 485-492
- [20] Montgomery J, Chen S. An analysis of the operation of differential evolution at high and low crossover rates [C]//IEEE Congress on Evolutionary Computation. 2010: 1-8
- [21] Liang J J, Qu B Y, Suganthan P N, et al. Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization [R]. Technical Report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China, Nanyang Technological University, Singapore, 2013
- [22] Derrac J, García S, Molina D, et al. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms [J]. Swarm and Evolutionary Computation, 2011, 1(1): 3-18