

## 差分进化算法参数控制与适应策略综述

杨振宇<sup>1</sup>, 唐珂<sup>2</sup>

(1. 华东师范大学 计算机科学技术系, 上海 200241; 2. 中国科学技术大学 计算机科学与技术学院, 安徽 合肥 230027)

**摘要:**差分进化算法逐渐成为进化计算领域最流行的随机搜索算法之一,已被成功用于求解各类应用问题. 差分进化算法参数设置与其性能密切相关,因此算法参数控制与适应策略设计是目前该领域的研究热点之一,目前已涌现出大量参数控制方案,但尚缺乏系统性的综述与分析. 首先简要介绍差分进化算法的基本原理与操作,然后将目前参数控制与适应策略分成基于经验的参数控制、参数随机化适应策略、基于统计学习的参数随机化适应策略和参数自适应策略4类进行系统性综述,重点介绍其中的参数适应与自适应策略. 此外,为分析各种参数控制与适应策略的功效,以实值函数优化为问题背景设计了相关实验,进一步分析各种策略的效率与实用性,实验结果表明,参数自适应控制策略是目前该领域最有效的方法之一.

**关键词:**进化计算; 差分进化; 参数控制; 适应策略; 自适应

**中图分类号:** TP18; O224 **文献标志码:** A **文章编号:** 1673-4785(2011)05-0415-09

## An overview of parameter control and adaptation strategies in differential evolution algorithm

YANG Zhenyu<sup>1</sup>, TANG Ke<sup>2</sup>

(1. Department of Computer Science and Technology, East China Normal University, Shanghai 200241, China; 2. School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China)

**Abstract:** Differential evolution algorithms have gradually become one of the most popular types of stochastic search algorithms in the area of evolutionary computation. They have been successfully applied to solve various problems in real-world applications. Since their performance often depends heavily on the parameter settings, the design of parameter control and adaptation strategies is one of the current hot topics of research in differential evolution. Although numerous parameter control schemes have been proposed, systematic overviews and analysis are still lacking. In this paper, first the basic principles and operations of the differential evolution algorithm were briefly introduced, and then a detailed overview was provided on different parameter control and adaptation strategies by dividing them into the following four classes: empirical parameter settings, randomized parameter adaptation strategies, randomized parameter adaptation strategies with statistical learning, and parameter self-adaptation strategies. The overview emphasized the latter two classes. To study the efficacy of these parameter control and adaptation strategies, experiments with the background of real-valued function optimization were conducted to compare their efficiency and practicability further. The results showed that the parameter self-adaptation is one of the most effective strategies so far.

**Keywords:** evolutionary computation; differential evolution; parameter control; adaptation strategies; self-adaptation

差分进化(differential evolution, DE)算法<sup>[1-3]</sup>于1995年由Storn和Price提出,是一种新颖的通过引入独特的差分变异模式进行迭代搜索的进化算法. 该方法因具有简单、易实现、高效、鲁棒性强等多种

优点<sup>[4]</sup>,已被广泛和成功地应用于全局优化、运筹管理、工程设计等领域<sup>[5-6]</sup>.

进化算法参数控制与适应一直都是进化计算领域的核心研究问题,这主要是因为该类方法的具体流程与操作通常由算法参数控制,使得这些参数的设置与算法性能息息相关<sup>[7]</sup>. 与传统进化算法类似,DE算法同样也具有其自身的控制参数. 经典

收稿日期:2010-11-02.

基金项目:海外及港澳学者合作研究基金资助项目(61028009).

通信作者:唐珂. E-mail: ketang@ustc.edu.cn.

DE 算法具有 3 个主要的控制参数<sup>[1]</sup>: 1) 种群大小  $N$ ; 2) 变异缩放因子  $F$ ; 3) 交叉概率  $P_{CR}$ . 除了种群大小  $N$  这个任何基于群体搜索的算法都具有的一般性参数外, 研究发现 DE 算法性能对另外 2 个参数的设置非常敏感, 而且往往与具体问题相关, 经验参数设置并不总能使该算法发挥出其最高性能, 而手动参数调节费时费力, 极大降低了算法的实用性. 为了提高 DE 算法的性能和实用性, 其参数的控制与适应策略设计已成为 DE 领域的热点研究方向, 除基于经验的参数设置外, 研究者们提出了大量更先进的参数控制与适应策略, 如参数随机化适应策略<sup>[8]</sup>、基于统计学习的参数随机化适应策略<sup>[9]</sup>、参数自适应策略<sup>[10]</sup>等.

针对 DE 算法的参数控制与适应问题, 目前研究者多以他们各自的视角提出了各种不同策略, 尚缺乏对这些现存策略的综述与系统性对比分析. 本文对目前研究中出现的主要的参数控制与适应策略进行综述, 将它们分成如下 4 类进行分析: 1) 基于经验的参数控制; 2) 参数随机化适应策略; 3) 基于统计学习的参数随机化适应策略; 4) 参数自适应策略. 除此之外, 还以实值函数优化问题为背景, 用具体实验结果对比来讨论各种参数控制与适应策略的实际效果.

## 1 差分进化算法

DE 的主要思想是引入一种全新的可利用当前群体中个体差异来构造变异个体的差分变异模式. 相对于传统进化算法, 差分变异模式是 DE 算法中最为独特的进化操作. 以经典 DE 算法为例, 在每代算法迭代过程中, 对于当前群体中的每个目标个体, 算法首先随机选择 2 个其他个体并使它们相减构成差分向量, 然后将该差分向量乘以一个缩放因子  $F$  后加到第 3 个随机个体上构成变异个体, 最后该变异个体再经过与对应目标个体的交叉和选择操作生成一个新个体进入下一代. 基于上述过程并结合文献<sup>[1]</sup>中的描述, 以下将以实值优化问题(即决策变量为实数)为背景, 具体介绍 DE 算法的群体表示与各种进化操作.

### 1.1 群体表示与初始化

对于实值优化问题, DE 算法中的群体一般表示成  $N$  个  $D$  维向量:

$$\{X_i \mid i = 1, 2, \dots, N\}.$$

式中: 实值向量  $X_i = (X_i(1), X_i(2), \dots, X_i(D))$  代

表群体中的一个个体,  $D$  是目标问题的决策变量个数, 也可以称为该问题的维数,  $N$  是群体大小. DE 算法的初始化方法与其他进化算法类似, 一般是在指定搜索空间内均匀随机生成  $N$  个  $D$  维实值向量构成其初始群体.

### 1.2 变异操作

给定当前群体  $\{X_i \mid i = 1, 2, \dots, N\}$ , 对其中的任意个体  $X_i$ , DE 算法的变异操作按照式(1)生成一个对应的变异个体.

$$V_i = X_{r_1} + F \times (X_{r_2} - X_{r_3}). \quad (1)$$

式中:  $X_{r_1}$ 、 $X_{r_2}$  和  $X_{r_3}$  是从当前群体中随机选择的 3 个互不相同的个体, 而且它们也不应与目标个体  $X_i$  相同; 缩放因子  $F$  是一个大于 0 的实常数, 实验表明它一般应满足条件  $F \in (0, 2]$ , 而  $F = 0.5$  通常是比较合适的设置<sup>[1]</sup>. 在具体算法实现中, 由于式(1)中的  $X_{r_2}$  和  $X_{r_3}$  是随机选择的 2 个个体,  $F$  取负值仅代表两者交换位置, 并不违背 DE 算法的设计原理, 因此 DE 研究中也会出现将  $F$  限定在区间  $[-2, 0) \cup (0, 2]$  的情况.

随着 DE 算法的发展, 根据差分向量构造方式的不同, 研究者还提出其他多种算法变种, 常用的有如下几种<sup>[2]</sup>:

$$V_i = X_{\text{best}} + F \times (X_{r_1} - X_{r_2}),$$

$$V_i = X_i + F \times (X_{\text{best}} - X_i) + F \times (X_{r_1} - X_{r_2}),$$

$$V_i = X_{\text{best}} + F \times (X_{r_1} - X_{r_2}) + F \times (X_{r_3} - X_{r_4}),$$

$$V_i = X_{r_1} + F \times (X_{r_2} - X_{r_3}) + F \times (X_{r_4} - X_{r_5}).$$

根据具体应用问题的不同, 这些变异模式各有优缺点, 但式(1)所表示的经典方式仍然是最常用、最有效的变异模式之一.

### 1.3 交叉操作

在完成变异操作后, DE 算法将在目标个体  $X_i$  和变异个体  $V_i$  之间执行一种离散交叉操作, 从而生成一个测试个体  $U_i$ , 该离散交叉可描述如下:

$$U_i(j) = \begin{cases} V_i(j), & R_j(0, 1) \leq P_{CR} \text{ or } j = j_{\text{rand}}; \\ X_i(j), & \text{otherwise.} \end{cases}$$

式中:  $R_j(0, 1)$  是一个在  $(0, 1)$  的均匀随机数发生器;  $j_{\text{rand}}$  是  $[1, D]$  的一个随机整数, 以确保不会出现测试个体  $U_i$  完全复制  $X_i$  的情况;  $P_{CR} \in [0, 1]$  是交叉概率, 用来控制在哪些决策变量上采用变异值, 一般可设置为 0.9<sup>[1]</sup>.

### 1.4 选择操作

对于每一个测试个体  $U_i$ , DE 算法采用如下的一对一的贪心选择方式:

$$X'_i = \begin{cases} U_i, & \text{if } f(U_i) \leq f(X_i); \\ X_i, & \text{otherwise.} \end{cases}$$

式中:  $f(\cdot)$  是目标函数(最小化问题);  $X'_i$  是代替  $X_i$  而进入下一代的子个体。

完成上述选择操作后, DE 算法得到一个新的群体  $\{X'_i | i=1, 2, \dots, N\}$  进入下一代, 从而可以迭代地继续执行进化搜索过程。

## 2 DE 算法参数控制与适应策略

### 2.1 基于经验的参数控制

在 DE 算法发展的早期, 其参数主要是根据人为经验设定为一些常数, 如文献[1]首先指出  $F=0.5$  是一个很好的初始选择, 如果群体出现早熟收敛现象, 则应该相应地增大  $F$  值, 但是小于 0.4 或大于 1 的  $F$  值往往只在极少数情况下才有用; 对于交叉概率  $P_{CR}$ , 该文献则指出 0.1 是一个合适的初始选择, 但考虑到大的  $P_{CR}$  值有助于提高算法收敛速度,  $P_{CR}=0.8$  通常也是不错的选择。

文献[11]以实值函数优化问题为背景对 DE 算法的参数展开了详细的数值分析, 指出了该算法性能对参数敏感的问题, 这些参数的设定不仅与具体问题相关, 而且它们之间也相互影响, 不易合理设置。对缩放因子  $F$ , 该文献推荐 0.6 为初始选择, 如果发现算法只能收敛到局部最优, 则应适当增大  $F$  值, 但当  $F>1$  时往往导致群体难以收敛; 对于交叉概率参数, 大的  $P_{CR}$  值有利于增加算法收敛速度, 但如果过大将有可能导致算法早熟, 介于 0.3~0.9 的值往往是比较好的设置。

在与其他算法的性能比较中, 文献[12]发现参数设置为  $F=0.5, P_{CR}=0.9$  的 DE 算法要显著优于粒子群优化 (particle swarm optimization, PSO) 算法<sup>[13]</sup> 及其改进版本 arPSO<sup>[14]</sup>, 也优于简单进化算法 (simple evolutionary algorithm, SEA)<sup>[15]</sup>。

也有研究者认为  $F$  应该设为稍大值以避免算法早熟, 如文献[16]中就建议将参数  $F$  和  $P_{CR}$  都设置为常数 0.9。对于 DE 算法的经验参数设置, 目前研究中一般都认为  $F=0.5, P_{CR}=0.9$  是比较有效的经典设置<sup>[17]</sup>。

### 2.2 参数随机化适应策略

虽然基于经验的参数控制可在一定程度上缓解 DE 算法的参数控制问题, 但是由于这种“经验”通常与具体目标问题密切相关, 不同的解空间分布往往具有不同的需求, 甚至同一问题的不同进化阶段

也可能具有不同需求, 不存在能适用于所有目标问题的统一固定参数设置, 所以经验参数控制逐渐不能满足 DE 算法在越来越广泛的应用问题上的求解性能要求。特别是对于特征未知的应用问题, 仍然需要通过费时费力的手动参数调节才能找到比较合适的经验参数, 这大大降低了 DE 算法的实用价值。为解决此问题, 研究者们逐渐开始考虑引入随机化适应策略进行参数控制。与将参数设置为固定常数的经验参数控制不同, 这种方法在每代对每个个体根据一个预先制定的概率分布模型生成伪随机数作为参数值, 其基本思想与进化算法思想类似, 主要是希望通过生成-测试的模式让算法自己选取比较合适的参数值。常用的概率分布模型有均匀分布 (uniform distribution)、高斯分布 (Gaussian distribution, 又称正态分布)、柯西分布 (Cauchy distribution) 等。

文献[8]将每次变异的缩放因子  $F$  设置为  $[0.5, 1]$  之间的均匀随机数, 提出一种改进的算法 DERSF; 文献[18]修改 DE 算法的变异模式提出 2 种改进算法 DERL 和 DELB, 它们也都使用了均匀随机数生成缩放因子  $F$ , 均匀随机数的范围限定为  $[-1, -0.4] \cup [0.4, 1]$ 。文献[19]提出一种 NSDE 算法, 将 DE 算法的交叉概率  $P_{CR}$  设置为  $[0, 1]$  之间的均匀随机数。实验分析发现这些改进算法都优于参数设置为  $F=0.5, P_{CR}=0.9$  的经典 DE 算法。

文献[20]针对多目标优化问题提出一种改进的 DE 算法 PDE, 该算法中缩放因子  $F$  被设置为均值为 0、标准差为 1 的高斯分布, 即  $F_i = N_i(0, 1)$ , 其中  $i$  表示对每次变异都重新生成一个  $F$  值。文献[21-22]通过引入多种变异策略提出一种自适应 DE 算法 SADE, 该算法中缩放因子  $F$  被设定为均值为 0.5、标准差为 0.3 的高斯分布, 即  $F_i = N_i(0.5, 0.3)$ , 其中参数 0.5 是参考了 DE 算法的经验参数设置, 目的是在比较好的经验值附近进行参数随机化, 给生成合适的参数值提供更大可能。

由 DE 变异公式(1)可以看出, 缩放因子  $F$  与算法搜索步长密切相关, 在不同的搜索阶段算法可能偏好不同的搜索步长, 比如当群体离全局最优点较远时, 较大的搜索步长将有助于算法快速收敛到好的子空间, 而当群体离全局最优点较近时, 小的搜索步长则有助于算法准确找到更优解<sup>[23]</sup>。缩放因子  $F$  的经验设置只能提供一种步长选择, 而随机化参数设置则可以提供多种选择, 这也是随机化参数设置通用性更强的原因。但是无论均匀分布还是高斯分

布,其搜索步长的广度都是有限的,当算法靠近多极值问题的局部最优点时,可能无法提供足够大的步长进行跳跃式搜索,因而仍然存在陷入局部最优的风险。

针对均匀分布和高斯分布的局限性,研究者们开始考虑引入范围更广的柯西分布进行参数随机化控制.文献[19]结合进化规划(evolutionary programming, EP)的特点,提出一种邻域控制差分进化算法 NSDE,该算法的缩放因子根据式(2)生成。

$$F_i = \begin{cases} N_i(0.5, 0.5), & \text{if } R_i(0, 1) \leq p; \\ C_i(0, 1), & \text{otherwise.} \end{cases} \quad (2)$$

式中: $p$  设为 0.5;  $N_i(0.5, 0.5)$  表示均值、标准差均为 0.5 的高斯随机数;  $C_i(0, 1)$  表示位置参数为 0、规模参数为 1 的柯西随机数;  $R_i(0, 1)$  表示  $(0, 1)$  的均匀随机数. NSDE 算法希望通过式(2)同时兼顾大小搜索步长,从而能有效搜索各种适应度分布特征未知的解空间。

文献[24]对 NSDE 算法进行了进一步改进,提出一种 SaNSDE 算法,其缩放因子  $F$  的控制方法与式(2)类似,只是根据经验将高斯随机数的参数修正为均值 0.5,标准差 0.3,并采用适应策略调整式(2)中的参数  $p$ . SaNSDE 算法中参数适应策略的有效性在很多测试函数上都得到了验证,并成功用于辅助求解问题规模高达 1 000 维的大规模实值优化问题<sup>[25]</sup>。

### 2.3 基于统计学习的参数随机化适应策略

参数随机化适应策略通过将 DE 算法的参数,即缩放因子  $F$  和交叉概率  $P_{CR}$ ,设置成随机数,增加了这些参数取值的多样性,从而使算法在面临无任何先验知识的问题时,仍然能够自动产生适合当前搜索需要的参数值,这在一定程度上提高了算法的性能.然而,这些用于控制 DE 算法参数的随机数发生器同样存在其自身的参数,比如高斯分布的均值和方差,这些参数同样需要合理设置.这些参数没有原算法的参数敏感,在上述参数随机化适应策略中一般都是根据经验设置,寻求这些参数的最优设置意味着存在进一步改进 DE 算法性能的空间,研究者们开始利用统计学习技术分析搜索过程中的历史信息,尝试从中找出规律以控制随机数发生器的参数。

文献[21-22]通过对交叉概率  $P_{CR}$  的分析指出,该参数在某个小范围内变动时对 DE 算法性能影响不大,但该范围不易确定.为解决此问题,该文献提出一种 SADE 算法,它假设交叉概率  $P_{CR}$  满足一个

均值为  $P_{CRM}$ 、标准差较小的高斯分布,算法开始时对每个个体根据式(3)生成一个  $P_{CR}$  值。

$$P_{CR_i} = N_i(P_{CRM}, 0.1). \quad (3)$$

式中: $P_{CRM}$  在算法开始时被设置为 0.5,在其周围生成的  $P_{CR_i}$  值将被使用若干代(如 SADE 算法中设置的 5 代),然后重新按照式(3)生成新的  $P_{CR_i}$ . 这样在每一代演化过程中,能使后代进入下一代的个体,对应的  $P_{CR_i}$  将会被记录在一个数组  $A_{CR}$  中,经过一定代数的累积后(如 SADE 算法中使用的 25 代),均值将按式(4)更新。

$$P_{CRM} = \frac{1}{A_{CR}} \sum_{k=1}^{|A_{CR}|} A_{CR}(k). \quad (4)$$

在每次  $P_{CRM}$  更新后,数组  $A_{CR}$  将会被清空进入下一次统计过程。

文献[24]分析认为当  $P_{CR}$  比较小时后代更易于进入下一代,但是该后代对应的适应度改进程度通常比较小;而大的  $P_{CR}$  值生成的后代虽然相对而言较难进入下一代,但当成功进入下一代时其引起的适应度改进程度往往要大很多,因此式(4)对应的适应策略存在将  $P_{CR}$  取较小值的导向.为缓解此问题,该文献提出一种加权的交叉概率适应策略,其基本原理与式(4)类似,区别在于每当在数组  $A_{CR}$  中记录成功  $P_{CR}$  值的时候,同时在另一数组  $A_{\Delta f} = \{\Delta f\}$  中额外记录对应个体的适应度改进值,即  $\Delta f = f(k) - f_{new}(k)$ ,然后  $P_{CRM}$  的更新公式修正为:

$$P_{CRM} = \sum_{k=1}^{|A_{CR}|} w_k A_{CR}(k), \quad (5)$$

$$w_k = \frac{A_{\Delta f}(k)}{(\sum_{k=1}^{|A_{\Delta f}|} A_{\Delta f}(k))}. \quad (6)$$

实验证明这种改进的参数适应机制使 DE 算法在一些复杂测试函数上的性能有了较显著的改进。

无论 SADE 还是 SaNSDE 都是采用一种离散的方式,通过历史信息更新用于参数控制的概率分布,这意味着只有当历史信息累积一定代数后才能被使用,当最大进化代数比较有限时,概率分布的参数被更新的频度将会很有限,这将影响参数适应性调整的及时性.文献[9]提出一种改进的 DE 算法 JADE,该方法采用了联系更新的方式适应性调整概率分布的参数.对于交叉概率  $P_{CR}$ ,JADE 仍然假设它服从均值为  $P_{CRM}$ 、标准差为 0.1 的高斯分布,即满足式(3)。

在每代进化过程中,算法将用  $S_{CR}$  记录能使对应后代进入下一代的  $P_{CR_i}$  值,并用式(7)更新  $P_{CRM}$ 。

$$P_{CRM} = (1 - c) \times P_{CRM} + c \times \text{mean}(S_{CR}). \quad (7)$$

式中:参数  $c$  是介于  $(0,1)$  的实数, JADE 中被设置为  $c=0.1$ . 对于缩放因子  $F$ , 该文献假设它服从位置参数为  $F_m$ , 规模参数为  $0.1$  的柯西分布:

$$F_i = C_i(F_m, 0.1). \quad (8)$$

式中:  $F_m$  被初始化为  $0.5$ , 如果  $F_i$  不在  $0 \sim 1$  之间, 将会被重新生成. 在每代进化过程中, 能使后代进入下一代的个体对应的  $F_i$  值将会被记录在数组  $S_F$  中, 然后根据式(9)更新.

$$F_m = (1 - c) \times F_m + c \times \text{mean}_L(S_F), \quad (9)$$

$$\text{mean}_L(S_F) = \frac{\sum_{F \in S_F} F^2}{\sum_{F \in S_F} F}. \quad (10)$$

式中:  $\text{mean}_L(\cdot)$  表示莱默均值 (Lehmer mean), 这里使用莱默均值而非一般均值是为了避免出现  $F$  过小而导致算法低效的情况.

文献[26]通过引入基于历史信息的可信度和影响适应度值变化程度的加权机制, 将上述2种适应策略扩展为控制任意与个体相关联的算法参数的一般性方法, 其基本思想是首先假设该参数的分布满足某个概率模型, 然后通过统计学习的方法逐渐调整概率模型的参数. 该文献使用这种一般性的参数随机化适应方法提出一种改进的 DE 算法 GaDE, 并将其成功用于求解大规模实值优化问题.

大量的实验结果表明, 基于统计学习的参数随机化适应策略不仅优于经验参数设置, 而且由于这类方法可以从历史信息中学习出一些参数变化趋势, 比简单的参数随机化适应策略更有方向性, 当这种方向性比较准确时, 这类方法将更有效.

#### 2.4 参数自适应策略

根据文献[27]的定义, 参数适应与自适应策略的本质区别在于后者须将算法参数放入个体编码中与决策变量一同进化. 依据这个条件, 无论是参数随机化策略还是基于统计学习的参数随机化策略都只能称之为适应策略, 而自适应策略一般被认为是进化算法参数控制的最高级手段, 在 DE 算法领域同样存在相关工作.

既然 DE 是一种非常有效的参数优化算法, 那么一种非常直观的自适应策略就是仍然采用 DE 这种模式来进化其自身的参数  $F$  和  $P_{CR}$ . 文献[28]提出一种 SPDE 算法, 该算法首先对每个个体都增加  $F_i$  和  $P_{CR_i}$  这2个参数作为决策变量放入个体表示, 在算法开始时  $F_i$  和  $P_{CR_i}$  被初始化为  $[0,1]$  的均匀随机数, 然后在每代进化运算时, 首先根据式(11)、

(12)生成.

$$F_i = F_{r1} + N_i(0,1) \times (F_{r2} + F_{r3}), \quad (11)$$

$$P_{CR_i} = P_{CR_{r1}} + N_i(0,1) \times (P_{CR_{r2}} - P_{CR_{r3}}). \quad (12)$$

式中:  $N_i(0,1)$  代表均值为  $0$ 、标准差为  $1$  的高斯随机数;  $r1$ 、 $r2$  和  $r3$  表示  $3$  个互不相同且也不与  $i$  相同的个体下标. 在这些参数进化完成后, 它们将参与到 DE 算法中的包含决策变量的个体的变异和交叉, 由于这些参数值存在于编码中, 如果每组参数对应生成的后代能成功进入下一代, 那么参数本身自然也进入了下一代, 从而达到一种控制参数层次的进化与自适应. 由于一般要求  $F$  和  $P_{CR}$  控制在区间  $[0,1]$  中, 因此如果式(11)、(12)生成的参数值超出了此范围, 将会应用修复规则使它们重新回到此范围. 文献[28]在一组多目标问题上测试了 SPDE 算法的性能, 与其他  $13$  种算法的比较表明了这种自适应策略比较有效. 类似地, 文献[29]建议用如下这种方式自适应控制参数  $F$ :

$$F_i = F_{r1} + N_i(0,0.5) \times (F_{r2} - F_{r3}). \quad (13)$$

而交叉概率  $P_{CR}$  被设置为根据均值为  $0.5$ 、标准差为  $0.15$  的高斯随机数  $N(0.5,0.15)$  生成. 文献[30]指出, 根据这种自适应策略设计的 SDE 算法要优于文献[28]中的 SPDE 算法.

文献[10]提出一种改进的 DE 算法 jDE, 它采用了另一种参数自适应策略. 算法 jDE 同样将  $F_i$  和  $P_{CR_i}$  与每个个体关联起来放入个体编码, 算法开始时  $F_i$  和  $P_{CR_i}$  被分别初始化  $0.5$  和  $0.9$ , 然后每代在决策变量进化前按式(14)、(15)更新.

$$F_i = \begin{cases} F_i + R_1 \times F_u, & \text{if } R_2 \leq \tau_1; \\ F_i, & \text{otherwise.} \end{cases} \quad (14)$$

$$P_{CR_i} = \begin{cases} R_3, & \text{if } R_4 \leq \tau_2; \\ P_{CR_i}, & \text{otherwise.} \end{cases} \quad (15)$$

式中:  $R_j (j \in \{1,2,3,4\})$  是  $[0,1]$  的均匀随机数;  $\tau_1$  和  $\tau_2$  分别表示更新参数  $F$  和  $P_{CR}$  的概率, 在 jDE 中被设置为  $\tau_1 = \tau_2 = 0.1$ ; 为了使新的  $F_i$  在  $[0.1,1.0]$  之间,  $F_l$  和  $F_u$  分别被设置为  $0.1$  和  $1.0$ . 文献[10]中设计了大量的数值实验分析了这种自适应策略的效果, 实验结果表明这种自适应策略在求解实值函数优化问题时非常有效, 使 jDE 成为目前最为高效的 DE 改进算法之一.

表1简要列出了本节介绍的各种 DE 算法变种及其采用的参数控制与适应策略.

表1 DE 算法变种及其对应的参数控制与适应策略

Table 1 DE variants and the corresponding parameter control and adaptation strategies

类别	算法	$F$	$P_{CR}$	文献出处
经验参数控制	DE	常数 0.5	常数 0.1 或 0.9	[1]
	DE	区间[0.6, 1]内常数	区间[0.3, 0.9]内常数	[11]
	DE	常数 0.5	常数 0.9	[12]
	DE	常数 0.9	常数 0.9	[16]
	ODE	常数 0.5	常数 0.9	[17]
参数随机化适应	DERSF	均匀随机数 $R(0.5, 1)$	常数 0.9	[8]
	SADE	高斯随机数 $N(0.5, 0.3)$	式(3)、(4)	[21]
	NSDE	式(2)	均匀随机数 $R(0, 1)$	[19]
	SaNSDE	式(2)	式(3)、(5)、(6)	[24]
	JADE	式(8)~(10)	式(3)、(7)	[9]
参数自适应	SPDE	式(11)	式(12)	[28]
	SDE	式(13)	高斯随机数 $N(0.5, 0.15)$	[29]
	jDE	式(14)	式(15)	[10]

### 3 实验分析

本文前面系统介绍了目前 DE 算法研究领域常用的参数控制与适应策略,为了对这些策略的实际效果有更直观的认识,笔者以实值函数优化为问题背景设计了一组实验进行验证.为保证实验结果比较的公平性,所有算法都采用经典 DE 变异公式(即式(1)),用表 2 来标记要测试的算法及其对应的参数控制策略.

表2 测试的 DE 算法变种及其参数控制与适应策略

Table 2 The tested DE variants and their parameter control and adaptation strategies

算法标识	$F$	$P_{CR}$
DE	常数 0.5	常数 0.9
DE <sup>U</sup>	随机数 $R(0, 1)$	随机数 $R(0, 1)$
DE <sup>G</sup>	随机数 $N(0.5, 0.3)$	随机数 $N(0.9, 0.1)$
DE <sup>C</sup>	随机数 $C(0.5, 0.1)$	随机数 $C(0.9, 0.1)$
SADE*	按 SADE 方式自适应	按 SADE 方式自适应
JADE*	按 JADE 方式自适应	按 JADE 方式自适应
jDE	按 jDE 方式自适应	按 jDE 方式自适应

由表 2 可以看出,第 1 种方法是采用经验参数设置的经典 DE 算法;接下来是 3 种采用参数随机化适应策略的算法,分别对应均匀分布、高斯分布和柯西分布;第 5、6 种算法采用基于统计学习的参数随机化适应策略,分别按文献[21]中的 SADE 和文献[9]中的 JADE 所采用的策略进行参数控制,由于这 2 种算法还采用了其他策略进一步改进算法性能,与这里只采用经典变异公式的方法稍有区别,分别用 SADE\* 和 JADE\* 来标记它们以示区别;最后一种采用参数自适应策略,与文献[10]中的 jDE 完全一致.

在测试集方面,采用了 13 个被广泛使用的经典实值函数优化函数<sup>[23]</sup>,表 3 给出了这些函数的定义与搜索空间.其中函数  $f_1 \sim f_4, f_6$  和  $f_7$  是单峰函数(即只有 1 个极值点),函数  $f_5, f_8 \sim f_{13}$  是多峰函数,而且极值点的个数非常多<sup>[23]</sup>,所有测试函数都被设定为 30 维,其最优值皆为 0.

在参数设置方面,所有算法的群体大小都设为 100,其参数  $F$  与  $P_{CR}$  的设置则按照表 2 及相关文献中的描述,对于比较复杂的函数  $f_3, f_4, f_5, f_8$  和  $f_9$  进化代数  $N$  设定为 5 000,而对于相对简单的其他函数,进化代数  $N$  设定为 1 500.对于每个测试函数,每个算法都独立运行 30 次统计实验结果,并计算均值与方差.表 4 和表 5 分别列出了算法在单峰和多峰函数上的均值与方差.

从表 4 的结果可以看出,基于经验参数设置的经典 DE 算法在函数  $f_1, f_3$  和  $f_6$  上具有不错的结果,但在其他函数上结果稍差.采用参数随机化适应策略的算法 DE<sup>U</sup>、DE<sup>G</sup> 和 DE<sup>C</sup> 一般比经典 DE 算法结果更好,而在函数  $f_4$  上结果却变得更差,说明参数随机化并不总是有效,这 3 个算法中使用高斯随机数的 DE<sup>G</sup> 效果最好.对于基于统计学习的参数随机化适应策略 SADE\* 和 JADE\*,它们在这些单峰函数上结果并不突出,仅在比较复杂的  $f_4$  上取得了更好结果.采用参数自适应的算法 jDE 几乎在所有函数上都取得了更好的结果,说明这种参数自适应策略对求解单峰函数比较有效.

从表 5 中算法在多峰函数上的结果看,经典 DE 算法在比较复杂的函数  $f_8$  和  $f_9$  上结果很差,在其他函数上结果一般.使用均匀随机数的 DE<sup>U</sup> 算法在大部分问题上都得到了更好结果,但在函数  $f_5$  上的结

果变得很差;使用高斯随机数的  $DE^G$  算法只在函数  $f_8 \sim f_{10}$  上得到了更好结果;使用柯西随机数的  $DE^C$  算法在函数  $f_5$  和  $f_{11}$  上结果不理想. 采用基于统计学习参数随机化适应策略的算法 SADE\* 在除  $f_5$  外的

函数上都取得了更好结果,而另一种 JADE\* 则在函数  $f_8$  和  $f_9$  上性能有显著提高. 对于自适应参数控制算法 jDE,它在除  $f_5$  外的函数上都取得了明显更好的结果.

表3 经典实值优化测试函数集

Table 3 Classical benchmark functions for real-valued optimization

测试函数	搜索空间
$f_1(x) = \sum_{i=1}^{30} x_i^2$	$[-100, 100]$
$f_2(x) = \sum_{i=1}^{30}  x_i  + \prod_{i=1}^{30}  x_i $	$[-10, 10]$
$f_3(x) = \sum_{i=1}^{30} (\sum_{j=1}^i x_j)^2$	$[-100, 100]$
$f_4(x) = \max_i \{  x_i , 1 \leq i \leq 30 \}$	$[-100, 100]$
$f_5(x) = \sum_{i=1}^{29} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]$
$f_6(x) = \sum_{i=1}^{30} (\lfloor x_i + 0.5 \rfloor)^2$	$[-100, 100]$
$f_7(x) = \sum_{i=1}^{30} ix_i^4 + \text{rand}[0, 1]$	$[-1.28, 1.28]$
$f_8(x) = -\sum_{i=1}^{30} (x_i \sin(\sqrt{ x_i })) + 30 \times 418.982\ 887\ 272\ 433\ 8$	$[-500, 500]$
$f_9(x) = \sum_{i=1}^{30} [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]$
$f_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{30} \sum_{i=1}^{30} x_i^2}\right) - \exp\left(\frac{1}{30} \sum_{i=1}^{30} \cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]$
$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^{30} x_i^2 - \prod_{i=1}^{30} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]$
$f_{12}(x) = \frac{\pi}{30} \{10 \sin^2(\pi y_1) + \sum_{i=1}^{29} \{(y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})]\} + (y_n - 1)^2\} + \sum_{i=1}^{30} u(x_i, 10, 100, 4)$	$[-50, 50]$
$f_{13}(x) = 0.1 \{ \sin^2(3\pi x) + \sum_{i=1}^{29} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2 [1 + \sin^2(2\pi x_{30})] \} + \sum_{i=1}^{30} u(x_i, 5, 100, 4)$	$[-50, 50]$

表4 采用各种不同参数控制与适应策略的 DE 算法在单峰函数上的结果

Table 4 Results of DE with different parameter control and adaptation strategies for unimodal functions

算法	参数	$f_1$ $N=1\ 500$	$f_2$ $N=1\ 500$	$f_3$ $N=5\ 000$	$f_4$ $N=5\ 000$	$f_6$ $N=1\ 500$	$f_7$ $N=1\ 500$
DE	均值	1.83e-13	1.19e-06	7.86e-11	2.91e-02	0.00e+00	9.67e-03
	方差	1.26e-13	7.19e-07	6.06e-11	7.33e-02	0.00e+00	2.78e-03
DE <sup>U</sup>	均值	3.13e-22	9.72e-14	3.99e-05	6.96e-02	0.00e+00	9.45e-03
	方差	2.14e-22	2.46e-14	7.00e-05	3.21e-01	0.00e+00	1.80e-03
DE <sup>G</sup>	均值	5.02e-24	3.61e-13	4.10e-15	7.37e+00	0.00e+00	5.92e-03
	方差	3.50e-24	2.25e-13	6.61e-15	2.97e+00	0.00e+00	1.53e-03
DE <sup>C</sup>	均值	3.03e-20	7.18e-11	1.07e-13	5.36e+00	0.00e+00	7.33e-03
	方差	2.91e-20	4.49e-11	1.25e-13	2.41e+00	0.00e+00	2.25e-03
SADE*	均值	7.34e-21	4.34e-13	3.55e+02	2.28e-05	0.00e+00	9.68e-03
	方差	3.83e-21	9.67e-14	5.11e+02	1.23e-04	0.00e+00	1.91e-03
JADE*	均值	4.09e-10	7.71e-06	5.65e+03	1.26e-02	0.00e+00	1.03e-02
	方差	6.07e-10	5.48e-06	2.49e+03	4.68e-02	0.00e+00	2.84e-03
jDE	均值	3.00e-28	5.13e-17	5.93e-14	2.19e-15	0.00e+00	7.03e-03
	方差	2.32e-28	2.85e-17	1.02e-13	1.45e-15	0.00e+00	2.00e-03

表5 采用各种不同参数控制与适应策略的DE算法在多峰函数上的结果

Table 5 Results of DE with different parameter control and adaptation strategies for multimodal functions

算法	参数	$f_5$	$f_8$	$f_9$	$f_{10}$	$f_{11}$	$f_{12}$	$f_{13}$
		$N=5\ 000$	$N=5\ 000$	$N=5\ 000$	$N=1\ 500$	$N=1\ 500$	$N=1\ 500$	$N=1\ 500$
DE	均值	1.49e-13	3.11e+03	9.05e+01	1.04e-07	3.64e-13	1.73e-14	7.42e-14
	方差	4.17e-13	1.88e+03	3.41e+01	2.95e-08	2.32e-13	1.32e-14	5.78e-14
DE <sup>U</sup>	均值	4.01e+00	3.95e+00	1.66e-01	4.19e-12	2.27e-23	2.37e-23	1.91e-22
	方差	3.27e+00	2.16e+01	3.77e-01	1.02e-12	7.26e-23	2.03e-23	1.16e-22
DE <sup>G</sup>	均值	8.63e-01	4.18e+02	8.95e+00	4.89e-13	2.47e-04	2.08e-02	1.32e-02
	方差	1.37e+00	2.46e+02	3.49e+00	2.36e-13	1.35e-03	9.60e-02	7.26e-02
DE <sup>C</sup>	均值	3.99e-01	4.13e+02	8.89e+00	4.51e-11	1.23e-03	2.05e-21	2.67e-19
	方差	1.22e+00	2.65e+02	2.76e+00	1.80e-11	3.98e-03	4.42e-21	1.36e-18
SADE*	均值	2.43e+01	3.95e+00	0.00e+00	2.22e-11	3.72e-22	2.13e-21	1.60e-20
	方差	1.06e+01	2.16e+01	0.00e+00	4.76e-12	4.60e-22	1.67e-21	7.23e-21
JADE*	均值	2.51e+01	2.65e-13	3.08e-33	7.79e-06	6.97e-08	3.57e-10	5.93e-10
	方差	1.92e+00	3.11e-14	6.00e-33	4.09e-06	1.39e-07	3.44e-10	6.75e-10
jDE	均值	1.26e-02	0.00e+00	0.00e+00	8.73e-15	7.68e-29	2.49e-29	2.03e-28
	方差	6.39e-02	0.00e+00	0.00e+00	2.15e-15	2.85e-28	2.33e-29	2.47e-28

从以上实验结果可以看出,采用经验参数控制的DE算法虽然能有效求解一部分问题,但由于其最优参数组合往往与问题相关而不易设置,导致这种方法一般不具有通用性;参数随机化适应策略和基于统计学习的参数随机化策略能弥补经验参数控制的不足,但并不是对任何问题都有效,精确度有待进一步提高;而参数自适应策略对绝大部分问题都适用,但也不排除对极少数问题存在更优的经验设置的可能(如函数 $f_5$ )。

#### 4 结束语

差分进化(DE)是进化计算领域一个非常重要的随机搜索算法新兴分支,在很多基准测试和实际应用问题上都表现出了卓越的性能,其新颖的进化操作起到了决定性的作用,而与这些操作相关的算法参数的控制与适应方法一直都是该领域的研究热点,目前已涌现出大量行之有效的策略.本文首先简要介绍了DE算法的基本原理与操作,然后着重综述了目前比较有代表性的参数适应策略,并加以分类总结,最后通过设计具体实验,对这些适应策略进行了对比分析,以验证它们的实际功效,相关实验结果表明,参数自适应控制策略是目前最有效的方法之一.此外,目前也存在很多通过设计更精巧的变异模式来改进DE算法性能的研究工作,将这些工作与参数自适应控制结合将是设计更高效DE算法的

有效研究手段之一。

#### 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, LAMPINEN J. Differential evolution: a practical approach to global optimization[M]. Berlin, Germany: Springer-Verlag, 2005.
- [3] CHAKRABORTY U K. Advances in differential evolution [M]. Berlin, Germany: Springer-Verlag, 2008.
- [4] STORN R, PRICE K. Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces[EB/OL]. [2010-10-25]. <http://www.icsi.berkeley.edu/~storn/TR-95-012.pdf>.
- [5] STORN R. System design by constraint adaptation and differential evolution[J]. IEEE Transactions on Evolutionary Computation, 1999, 3(1): 22-34.
- [6] THOMSEN R. Flexible ligand docking using differential evolution[C]//Proceedings of the 2003 IEEE Congress on Evolutionary Computation. Canberra, Australia, 2003: 2354-2361.
- [7] BACK T, SCHWEFEL H P. An overview of evolutionary algorithms for parameter optimization[J]. Evolutionary Computation, 1993, 1(1): 1-23.
- [8] DAS S, KONAR A, CHAKRABORTY U K. Two improved differential evolution schemes for faster global search[C]//



- Proceedings of the 2005 Genetic and Evolutionary Computation Conference. Washington, DC, USA, 2005: 991-998.
- [9] ZHANG J, SANDERSON A C. JADE: adaptive differential evolution with optimal external archive[J]. IEEE Transactions on Evolutionary Computation, 2009, 13(5): 945-958.
- [10] BREST J, GREINER S, BOSKOVC 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.
- [11] GAMPERLE R, MULLER S, KOUMOUTSAKOS P. A parameter study for differential evolution[C]//WSEAS International Conference on Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation. Interlaken, Switzerland, 2002: 293-298.
- [12] VESTERSTROM J, THOMSEN R. A comparative study of differential evolution, particle swarm optimization and evolutionary algorithms on numerical benchmark problems[C]//Proceedings of the 2004 IEEE Congress on Evolutionary Computation. Portland, USA, 2004: 1980-1987.
- [13] KENNEDY J, EBERHART R C. Particle swarm optimization[C]//IEEE International Conference on Neural Networks. Perth, Australia, 1995: 1942-1948.
- [14] RIGET J, VESTERSTROM J. A diversity-guided particle swarm optimizer—the arPSO, Technical Report 2002-02[R]. Aarhus, Denmark: Department of Computer Science, University of Aarhus, 2002.
- [15] THOMSEN R. Flexible ligand docking using evolutionary algorithms: investigating the effects of variation operators and local search hybrids[J]. BioSystems, 2003, 72(1/2): 57-73.
- [16] ROMKKONEN J, KUKKONEN S, PRICE K V. Real-parameter optimization with differential evolution[C]//Proceedings of the 2005 IEEE Congress on Evolutionary Computation. Edinburgh, UK, 2005: 506-513.
- [17] RAHNAMAYAN S, TIZHOOSH H R, SALAMA M A. Opposition-based differential evolution[J]. IEEE Transactions on Evolutionary Computation, 2008, 12(1): 64-79.
- [18] 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.
- [19] YANG Zhenyu, YAO Xin, HE Jingsong. Making a difference to differential evolution[M]//MICHALEWICZ Z, SIARRY P. Advances in Metaheuristics for Hard Optimization. Berlin, Germany: Springer, 2008: 397-414.
- [20] ABBASS H A, SARKER R, NEWTON C. PDE: a Pareto-frontier differential evolution approach for multi-objective optimization problems[C]//Proceedings of the 2003 IEEE Congress on Evolutionary Computation. Canberra, Australia, 2003: 971-978.
- [21] QIN A K, SUGANTHAN P N. Self-adaptive differential evolution algorithm for numerical optimization[C]//Proceedings of the 2005 IEEE Congress on Evolutionary Computation. Edinburgh, UK, 2005: 1785-1791.
- [22] 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.
- [23] YAO Xin, LIU Yong, LIN Guangming. Evolutionary programming made faster[J]. IEEE Transactions on Evolutionary Computation, 1999, 3(2): 82-102.
- [24] YANG Zhenyu, TANG Ke, YAO Xin. Self-adaptive differential evolution with neighborhood search[C]//Proceedings of the 2008 IEEE Congress on Evolutionary Computation. Hong Kong, China, 2008: 1110-1116.
- [25] YANG Zhenyu, TANG Ke, YAO Xin. Large scale evolutionary optimization using cooperative coevolution[J]. Information Sciences, 2008, 178(15): 2985-2999.
- [26] YANG Zhenyu, TANG Ke, YAO Xin. Scalability of generalized adaptive differential evolution for large-scale continuous optimization[J]. Soft Computing, 2011 (in press).
- [27] HINTERDING R, MICHALEWICZ Z, EIBEN A E. Adaptation in evolutionary computation: a survey[C]//Proceedings of the 1997 IEEE International Conference on Evolutionary Computation. Indianapolis, USA, 1997: 65-69.
- [28] ABBASS H A. The self-adaptive Pareto differential evolution algorithm[C]//Proceedings of the 2002 IEEE Congress on Evolutionary Computation. Honolulu, USA, 2002: 831-836.
- [29] OMRAN M, SALMAN A, ENGELBRECHT A. Self-adaptive differential evolution[C]//Proceedings of the International Conference on Computational Intelligence and Security. Xi'an, China, 2005: 192-199.
- [30] SALMAN A, ENGELBRECHT A, OMRAN M. Empirical analysis of self-adaptive differential evolution[J]. European Journal of Operational Research, 2007, 183(2): 785-804.

#### 作者简介:



杨振宇,男,1982年生,讲师,博士,主要研究方向为进化计算与应用,发表学术论文多篇,其中被SCI、EI检索10余篇。



唐珂,男,1981年生,教授,博士,IEEE Computational Intelligence Magazine副编辑,IEEE 计算智能协会 Emergent Technologies Technical Committee 委员。主要研究方向为计算智能、机器学习、模式识别等,发表学术论文50余篇。