

Optimization of hybrid evolutionary strategy algorithm

Zhu Jun

Ji'nan University, Guangzhou

Abstract: The Gaussian and Cauchy mutation operators were combined to achieve the dynamic balance between global exploration and local search. The improved evolutionary strategy also USES recombination operator, constraint treatment and elite retention strategy to further improve the performance of the algorithm. The improved hybrid strategy algorithm is applied to solve the extreme value problem of multi-peak function. The numerical simulation results show the effectiveness of the algorithm.

Key words: Evolutionary algorithms; Evolutionary strategy; The function extreme value

Received: 2019-09-25; Accepted: 2019-11-01; Published: 2019-11-04

混合进化策略算法优化探析

朱 俊

暨南大学, 广州

邮箱: jzhu117@qq.com

摘 要: 采用基于个体排序的随机自适应 Gaussian-Cauchy 混合变异策略, 将 Gaussian 和 Cauchy 变异算子结合起来以达到全局探索和局部搜索之间的动态平衡。改进的进化策略还使用重组算子、约束条件处理、精英保留策略以进一步提高算法的性能。将改进的混合策略算法应用于求多峰值函数极值问题, 数值仿真实验结果显示了该算法的有效性。

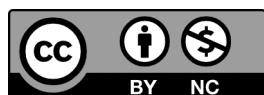
关键词: 进化算法; 进化策略; 函数极值

收稿日期: 2019-09-25; 录用日期: 2019-11-01; 发表日期: 2019-11-04

Copyright © 2019 by author(s) and SciScan Publishing Limited

This article is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/).

<https://creativecommons.org/licenses/by-nc/4.0/>



现有数学基础所提出的各种解决问题的方法, 存在鲁棒性差, 难以取得全局优化的缺点, 而生物进化的过程有极强的鲁棒性及全局优化特点。进化算法

是模拟自然界生物进化过程与机制求解优化与搜索问题的一类自组织、自适应人工智能技术，是一种宏观意义下的仿生优化算法。它模仿的是一切生命与智能的生成与进化过程。它不仅模拟达尔文“优胜劣汰、适者生存”的进化原理激励好的结构，而且也通过模拟孟德尔等人的遗传变异理论在优化过程保持已有的结构，同时寻找更好的结构。

模拟生物进化，产生三种典型的优化理论模型：1) 遗传算法 GA (Gen Algorithm)；2) 进化策略 ES (Evolution Strategy)；3) 进化规划 EP (Evolutionary Programming)。这些方法很相似，但各有不同的侧重，并且在不同背景下独立发展出来。进化策略 (Evolution Strategy, ES) 是由德国数学家 I.Rechenberg 和 H.P.Schwefel 等于 20 世纪 60 年代提出的一类数值优化算法。本文针对多峰值函数问题求解极值进行优化。通常采用的 Gaussian 变异算子求解函数极值，具有较好的局部寻优能力，但是最终求出来的极值有时可能只是局部的一个极值而非全局的极值。而 Cauchy 变异算子具有全局的寻优能力和局部逃逸能力。因此，综合使用这两种方式的优点，组成一种混合的变异算子以优化算法，能够达到局部和全局优化平衡。并使用约束条件处理、精英保留策略等提高算法的性能。

1 进化策略

进化策略是进化算法中具有自适应机制的一类数值优化算法 [1]。它使用随机变量遗传编码，即用 $2n$ 维实数组 (X, σ) 来编码实变量 $X \in R_n$, $X = (x_1, x_2, \dots, x_n)^T$, $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_n)^T$, 而解码变换为 $A = (X, \sigma)X + N(0, \sigma)X$, 这里 $N(0, \sigma)$ 表示各分量相互独立，均值为 0，以 $\text{diag}\{\sigma_1^2, \dots, \sigma_n^2\}$ 为协方差的正态分布随机向量。ES 的个体空间 $H_L = R^n \times (R_0^+)^n R^{2n}$, 而繁殖仅由变异操作完成。进化策略的具体算法步骤如下 [1]：

Step 1 (初始化) 设定进化终止准则并置时间 $t := 0$ ；随机生成 μ 个个体 $(X^{(i)}, \sigma^{(i)})$ ($i=1, 2, \dots, \mu$) 组成初始种群 $X(0) = \{(X^{(i)}, \sigma^{(i)}) | i=1, 2, \dots, \mu\}$ 。

Step 2 (种群进化) 包括变异和选择两个过程。变异。随机从 $X(t)$ 中选取 λ ($\lambda > \mu$) 个个体执行变异操作，生成中间种群 $X'(t) = \{(X(j), \sigma(j))$

$l_j=1, 2, \dots, \lambda\}$ 。

1) 选择。从 $X(t) \cup X'(t)$ 中依适应度大小选择出 μ 个个体作为新一代种群 $X(t+1)$ 。

Step 3 (终止检验) 如果满足进化终止准则, 则输出 $X(t+1)$ 中适应度最高的个体作为最优解, 终止计算; 否则, 置 $t:=t+1$, 并转 Step 2。

在上述 ES 中, 变异是繁殖新个体的唯一操作, 有些 ES 实现中有重组或交叉操作。假设 $X(t)$ 中某一个体 $A=(x, \sigma)$ 经过变异得到一个新的个体 $A'=(x', \sigma')$, 则新个体 A' 中的 $X'=(x'_1, x'_2, \dots, x'_n)^T$ 和 $\sigma'=(\sigma'_1, \sigma'_2, \dots, \sigma'_n)^T$ 分别定义为

$$\begin{cases} x'_i = x_i + N(0, \sigma'_i), \\ \sigma'_i = \sigma_i \exp[\tau N(0, 1) + \tau' N(0, \sigma_i)] \end{cases} \quad (1)$$

$\sigma_i = \sigma_i \exp[\tau N(0, 1) + \tau' N(0, \sigma_i)]$ 。其中 $N(0, \sigma_i)$ 表示均值为 0, 方差为 σ_i 的正态分布随机变量; τ 和 τ' 是表示整体步长与分量步长的两个参数。进化策略中的选择操作有两种典型方式: 一是父代种群 $X(t)$ 不参与竞争, 而直接从中间种群 $X'(t)$ 中选择 μ 个适应度高的个体作为新一代种群; 二是父代种群参与竞争, $X(t+1)$ 中的 μ 个个体是从 $X(t) \cup X'(t)$ 中的 $\lambda + \mu$ 个个体所选择出的 μ 个优秀个体。按照习惯, 以第一种方式执行选择出的 ES 称为 (μ, λ) -ES, 而以第二种方式执行选择的 ES 称为 $(\mu + \lambda)$ -ES。本文选用第二种方式。

2 改进的进化策略

2.1 混合进化策略

Yao 等用满足 Cauchy 分布的 Cauchy 变异算子代替通常采用的 Gaussian 变异算子, 提出了快速进化策略 [2] 和快速进化规划 [3]。由于 Cauchy 变异算子产生大变异的概率高, 可以减少陷入局部极值的危险, 而 Gaussian 变异算子则有良好的局部搜索能力, 可以加快局部收敛速度, 因而提出了一种混合变异, 将 Cauchy 变异算子和 Gaussian 变异算子的优点结合起来 [2] [3]。具体方法

是：按照目标函数值对父代个体降序排序 [4]，据此排序为每个父代个体分别制定 Gaussian 变异和 Cauchy 变异选择概率，使得排名越靠前的个体 Gaussian 变异概率大而 Cauchy 变异的概率小，排名靠后的个体 Cauchy 变异的概率越大而 Gaussian 变异的概率越小，从而将 Gaussian 变异的良好局部搜索能力和 Cauchy 变异的大范围搜索能力有效结合起来，达到全局搜索和局部搜索之间的一种动态平衡 [5]。

2.2 重组算子

进化策略使用变异算子作为主要搜索算子，适当的算子组合可以提高进化策略的性能，是进化的辅助手段。为此在改进进化策略中引入策略参数的

中间重组，即修改公式 (1) 的第二式为

$$\sigma'_i = \text{rec}(\sigma_i, \sigma_{mj}) \exp[\tau N(0, 1) + \tau' N(0, \sigma_i)] \quad (2)$$

其中重组算子 $\text{rec}(\cdot, \cdot)$ 定义为： $\text{rec}(\sigma_i, \sigma_{mj}) = \sigma_i + (1 - \beta_j) \sigma_{mj}$ ，随机整数 $m_j \in \{1, 2, \dots, u\}$ 且 $m_j \neq i$ 。

β_j 为 $[0, 1]$ 区间内均匀分布的随机数。对于下标 j 的每个值， m_j 和 β_j 都要重新抽样。在传统进化策略中，随着进化代数的增加，策略参数逐渐趋向零，最终导致进化基本停止，为此在公式 (2) 的基础上对策略参数的最小值进行限制： $\sigma'_i = \max[\zeta, \sigma'_i]$ ，其中 ζ 为预先设定的策略参数下界。

2.3 约束条件处理

在结构优化问题中，经常遇到约束条件很复杂的情况。一般将约束条件分为三类 [5]：区域约束，不等式约束和等式约束。区域约束在初始化的时候加以处理，实验搜索空间的点与可行的点一一对应。对不等式约束和等式约束采用罚函数方法，将违反约束的解施加惩罚，使其适应度大大降低，减少在下一代的存活几率。具体形式如下式 [5]：

$$\begin{cases} F(X) = f(x) + \sum_{j=1}^J R_j(g'_j(X))^2, \\ g'_j(X) = \max(0, g_j(X)) \end{cases} \quad (3)$$

式中 $F(X)$ 为惩罚后目标函数； $f(X)$ 为目标函数； $g_j(X)$ 为第 j 个约束函数；

R_j 为惩罚因子; $g'_j(X)$ 为约束违法函数; J 为所有约束函数个数。

2.4 精英保留策略

保留目前存在的最优个体, 保证其不被交叉和变异等遗传算子破坏, 它是遗传算法收敛性的一个重要保证条件。基于 Markov 链的数学理论表明, 保留最优个体策略的遗传算法能够以概率 1 收敛于最优解。具体操作为: 计算新群体的适应值, 若新群体中的最优个体优于上代保存的最优个体, 则用该个体替换上代保留的最优个体; 否则, 用上代的最优个体替换新群体中最差的个体 [4]。

3 改进的策略用于求解多峰值函数极值

在工程技术和自然科学领域, 许多现象的定量分析及优化改进往往可以抽象的归结为求函数的极值问题。所以研究函数极值问题的求解有着非常重要的意义。本文选择有代表性的函数 $f(x_1, x_2) = a * (x_1^b - x_2)^c + (x_1 - x_2)^d$, ($e \leq x_1 \leq f, g \leq x_2 \leq h$)。取 $a=50, b=c=d=-e=f=2, -g=h=3$ 。用改进的进化策略算法求出极小值。

依次迭代, 得到每次迭代过程中的进化结果值和误差值。根据算法实现后得到的数据, 进化曲线图和进化误差曲线图如下:

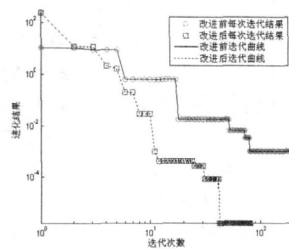


图 1 改进前后进化曲线图

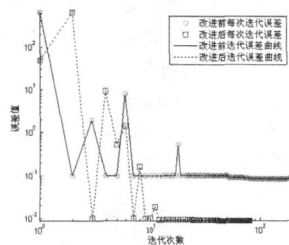


图 2 改进前后误差曲线图

表 1 改进前后结果对照表

| 改进前后 | 次序 | 极小值 | 进化代数 | 学习时间 / s | 误差 |
|------|----|-----------------------|------|----------|--------------|
| 改进前 | 1 | 0.000 433 126 | 200 | 0.53 | 0.081 746 5 |
| | 2 | 0.016 537 300 | 47 | 0.12 | 0.004 241 8 |
| | 3 | $1.443\ 55e^{-0.005}$ | 200 | 0.63 | 0.024 256 0 |
| 改进后 | 1 | 0.001 297 570 | 13 | 0.04 | 0.000 693 81 |
| | 2 | $5.081\ 07e^{-0.005}$ | 80 | 0.19 | 0.005 616 56 |
| | 3 | $1.680\ 09e^{-0.006}$ | 80 | 0.41 | 0.008 961 09 |

通过实验结果可以得到以上结果对照表,从中可以看出改进后,极小值更快速趋近于 0,进化代数减少,学习时间也减少很多,误差比改进前也更接近于 0。如果增加迭代次数,将会使误差更小,而且结果更趋于 0,因此,改进后效果很明显。

参考文献

- [1] FOGEL D B. An Introduction to simulated evolutionary optimization [J]. IEEE Trans. On Neural Network, 1994, 5 (1): 3-14.
- [2] Yao X, Liu Y. Fast evolution strategies [J]. Control and Cybernetics, 1997, 26 (3): 467-496.
- [3] Yao X, Liu Y, Lin G M. Evolutionary programming made faster [J]. IEEE Trans Evolutionary Computation, 1999, 3 (2): 82-102.
- [4] 杨春松, 程文明. 一种进化类混合算法的研究 [J]. 计算机仿真, 2007, 24 (10): 169-172.
- [5] 柯晶, 姜静, 李歧强. 改进进化策略及其在神经网络训练中的应用 [J]. 计算机工程与应用, 2006 (4): 68-70.
- [6] Ohkura K, Malsumura Y, Ueda K. Robust evolution strategies [J]. Applied intelligence, 2001, 15 (3): 153-169.
- [7] 谢红薇, 田超, 余雪丽. 基于双基因变异方式的混合进化策略 [J]. 微计算机信息, 2008 (6): 230-232