

基于混合优化策略的微分进化改进算法

赵光权, 彭喜元, 孙 宁

(哈尔滨工业大学自动化测试与控制系, 黑龙江哈尔滨 150080)

摘 要: 微分进化算法具有控制参数少、鲁棒性强、易于使用等优点, 并具有不同的优化策略. 本文在对微分进化算法各优化策略性能进行分析的基础上, 提出了基于混合优化策略的微分进化改进算法. 改进算法的主要思想是将种群中的个体随机地分成两组, 每组采用不同的优化策略. 利用五个标准的优化算法测试函数对改进算法的收敛速度和搜索成功率进行了测试, 并与动态微分进化算法和微粒群算法进行了比较. 实验结果表明, 本文提出的改进算法在保证算法搜索成功率的同时, 大大提高了算法搜索效率.

关键词: 优化算法; 优化策略; 微分进化算法

中图分类号: TP18 **文献标识码:** A **文章编号:** 0372-2112 (2006) 12A-2402-04

A Modified Differential Evolution Algorithm with Hybrid Optimization Strategy

ZHAO Guang-quan, PENG Xi-yuan, SUN Ning

(Department of Automatic Test and Control, Harbin Institute of Technology, Harbin, Heilongjiang 150080, China)

Abstract: The differential evolution algorithm is robust, easy to use, requires few control parameters, and has various optimization strategies. Based on analysis of advantages and disadvantages of these optimization strategies, a modified differential evolution algorithm with hybrid optimization strategy is proposed. The main idea of the modified differential evolution algorithm is to divide all of the individuals into two groups randomly, and the two groups adopt different optimization strategies. The convergence speed and search succeed probability of the modified differential evolution are tested using five benchmark functions for optimization algorithm, and the results are compared with dynamic differential evolution and particle swarm optimization. From the simulation results, it is observed that the search efficiency of the modified differential evolution is significantly improved as well as the high search succeed probability is ensured.

Key words: optimization algorithm; optimization strategy; differential evolution algorithm

1 引言

全局优化问题大量存在于计算机科学、管理科学、控制工程等科研领域, 因此成为科学研究人员长期的研究热点. 以遗传算法 (GA)、模拟退火 (SA) 和微粒群算法 (PSO) 为代表的智能优化算法为优化问题的解决提供了新的途径, 这些算法模拟自然界的一些自然现象, 形成进化的流程和规则, 使计算机可以模仿自然界的进化过程进行寻优计算以解决科学研究中的优化问题.

1995 年 Rainer Storn 和 Kenneth Price 提出的微分进化算法 (differential evolution algorithm, 以下简称 DE 算法) 是一种实数编码的基于种群进化的全局优化算法, 已被证明在求优过程中具有高效性、收敛性、鲁棒性等优点^[1]. DE 算法已在滤波器设计、神经网络参数训练、聚类分析、机器人路径规划等工程领域得到应用并取得了良好的效果^[2-5]. DE 基本算法的核心思想是利用随机偏差扰动产生新的中间个体. 其产生中间个体的方式决定了 DE 算法在许多问题上都有很好的收敛表现, 但搜索速度则相对缓慢. 如果优化问题是计算成本很高即每计算一次目

标函数值都需要很长时间的问题, 那么过多次数的计算目标函数值就会使得算法不可行, 因此减少算法收敛所需目标函数评价次数和收敛时间就具有很强的现实意义.

最新的 DE 改进算法有 Anyong Qing 提出的动态微分进化算法 (简称 DDE), 其基本思想是对种群中的个体进行动态更新, 而不是等所有个体都更新完毕后再以新种群替代旧种群^[6]. Paul K Bergey 提出了 MDE 算法, MDE 将种群中的个体按适应度排序, 适应度更高的个体以较大的概率被选中作为变异操作的父体^[7]. Jianyong Sun 等将 DE 算法和 EDA 算法进行了混合, 提出了 DE/EDA 混合算法^[8]. P Kaelo 等提出了 DERL 算法和 DELB 算法, 其中, DERL 算法主要是对参与变异操作的三个个体按适应度值高低排序, 并将适应度最高的个体作为基点向量, DELB 算法则对算法的选择操作进行了改进, 使种群中的个体在每次迭代过程中有多次的改善机会^[9]. 这些改进算法都不同程度地提高了 DE 算法的性能. 应该说, DE 算法在搜索成功率性能方面已非常出色, 但在搜索速度上仍有提高的空间. 在实际应用中, DE 算法可以采用不同的优化策略, 本文在对 DE 各优化策略进行分析的基础上提出了

基于混合优化策略的微分进化改进算法.

2 DE 基本算法

DE 算法是求解有 n 个连续变量全局优化问题的算法. 全局优化问题都可以转化为求解如下的最小化问题:

$$\min f(X) \quad (1)$$

其中 $X = (x_1, x_2, \dots, x_n)$ $D \subset R^n$ 是连续变量, 目标函数 $f: D \rightarrow R$ 可能是不可微的.

同所有的进化优化算法一样, DE 算法也是对候选解的种群进行操作. DE 利用实数值参数向量作为每一代的种群, DE 的中间个体是通过把种群中两个个体之间的加权差向量加到第三个个体上来产生的, 这称为“变异”; 然后将中间个体的参数与当前个体的参数按照一定规则混合来产生所谓的候选个体, 通常称为“交叉”; 如果候选个体的目标函数值小于当前个体的目标函数值, 候选个体就在下一代中代替当前个体, 这一操作称为“选择”. 种群中所有个体都要被作为当前个体进行一次变异、交叉和选择操作. 这种利用随机偏差扰动产生新个体的方式可以获得一个具有非常好收敛性质的自适应程序^[10]. DE 基本算法的流程以及变异、交叉、选择算子的详细描述请见参考文献[1], 此处不再赘述.

DE 算法中有三个控制参数: F 、 CR 、 NP . 其中 F 和 CR 影响搜索过程的鲁棒性和算法的收敛速度. 种群规模 NP 取值一般为 $2D \sim 20D$, 且至少应大于 4. F 取值范围一般为 $0.4 \sim 1.0$, 建议初始值选为 0.5. CR 初始值建议选为 0.5.

DE 算法有几种变形, 即有不同的优化策略. 为了区分不同的 DE 算法, 通常将 DE 表示为: $DE/x/y/z$. 其中 x 表示变异操作中的基点向量是随机选取的还是选择当前种群中的最优个体, y 表示差分向量的个数, z 表示交叉方式, 主要有指数交叉和二项式交叉两种. 一般来说, DE 基本算法表示为 $DE/rand/1/bin$, 该优化策略在实际中应用最为广泛.

3 改进算法 HDE

3.1 DE 算法各优化策略比较

为了更清楚地了解各种优化策略下 DE 算法的性能, 本文利用各优化策略对 2 维 Schaffer 函数进行了测试, 并将搜索成功率、目标函数评价次数 (number of function evaluations, 以下简称为 nfe) 和收敛时间 (t) 列于表 1. Schaffer 函数的相关描述和 DE 算法的参数设置见本文第 4 节.

表 1 DE 各优化策略对 Schaffer 函数的测试结果

算法	算法描述	成功率	nfe	t/s
DE1	best/1/exp	70%	3094	0.0187
DE2	rand/1/exp	100%	12330	0.0760
DE3	rand-to-best/1/exp	100%	12294	0.0755
DE4	best/2/exp	100%	15912	0.0983
DE5	rand/2/exp	100%	42036	0.2576
DE6	best/1/bin	70%	3248	0.0212
DE7	rand/1/bin	100%	12873	0.0796
DE8	rand-to-best/1/bin	60%	11448	0.0720
DE9	best/2/bin	90%	17706	0.1109
DE10	rand/2/bin	100%	35406	0.2220

3.2 HDE 算法原理

由表 1 可见, 不同的优化策略具有不同的搜索性能. 通过目标函数评价次数和收敛时间可以看出算法的搜索效率, 通过成功率可以看出算法的全局收敛性能. 一般来说, 如果变异操作中的基点向量是随机抽取的, 则算法全局收敛性好, 不易陷入局部最优, 但收敛所需目标函数评价次数和收敛时间也偏大; 如果变异操作的基点向量选择了当前种群中的最优个体, 则收敛速度相对较快, 但搜索成功率则不高, 较易陷入局部最优. 例如 DE6 算法中变异操作的基点向量选择了当前种群中的最优个体, 使其收敛速度相对较快, 但搜索成功率仅为 70%. 又如 DE7 算法中变异操作的基点向量是随机选取的, 因此其全局收敛性好, 搜索成功率为 100%, 但其收敛速度则较 DE6 缓慢. 如果将这两种优化策略进行混合, 取长补短, 可以预见算法的性能会有所改善, 即在保证算法良好收敛性的同时, 加快算法的收敛速度.

基于以上思想, 本文提出了基于混合优化策略的微分进化改进算法 (Hybrid Differential Evolution Algorithm, 以下简称为 HDE). HDE 算法可以有两种形式, 分别记为 HDE1、HDE2. HDE1 算法的流程图如图 1 所示, 即如果按照 DE7 优化策略所得候选个体的适应度相对于当前个体没有得到改善, 则利用 DE6 优化策略获得一个新的候选个体, 并重新进行选择操作. HDE2 算法则在每次迭代过程中将种群中的个体随机地分成两组, 每组采用不同的优化策略进行优化, 比如种群中 50% 个体采用 DE7 优化策略获得候选个体, 另 50% 个体则采用 DE6 优化策略获得候选个体, 选择操作不做任何改变, 本文所讨论的 HDE2 算法即采用 50% 比例进行混合优化.

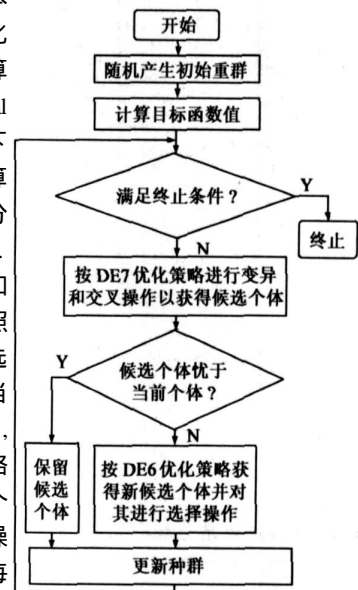


图 1 HDE1 算法流程图

4 HDE 算法仿真实验

4.1 测试函数

为了检验 HDE 算法的性能, 选择了 5 个常用的优化算法测试函数对算法进行测试, 它们分别是 sphere 函数 (f_1)、Rosenbrock 函数 (f_2)、Rastrigin 函数 (f_3)、Griewank 函数 (f_4) 和 Schaffer 函数 (f_5)^[11]. 这 5 个函数的表达式分别见式 (2) ~ 式 (6), 测试停止迭代条件以及 2 维图形如表 2 所示. 除了 Schaffer 是 2 维函数外, 其他函数都用 30 维变量进行测试. 5 个函数中前两个为单峰函数, 只有一个极小值. 后三个为多峰函数, 有多个局部极小值, 其中 Rastrigin 和 Griewank 函数有很大

的曲率,可以引导搜索到全局极小,而 Schaffer 函数在全局极小附近剧烈震荡,一般算法难以得到最优解.

$$f_1(\bar{x}) = \sum_{i=1}^n x_i^2 \quad (2)$$

$$f_2(\bar{x}) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2) \quad (3)$$

$$f_3(\bar{x}) = \sum_{i=1}^n (x_i^2 - 10\cos(2x_i) + 10) \quad (4)$$

$$f_4(\bar{x}) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \sum_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (5)$$

$$f_5(\bar{x}) = 0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2} \quad (6)$$

表 2 测试函数描述

函数	维数 n	范围 [x _{min} , x _{max}]	最优值 f	停止条件	2 维图形
f ₁	30	[-100, 100] ^a	0	0.01	
f ₂	30	[-30, 30] ^a	0	100	
f ₃	30	[-5.12, 5.12] ^a	0	100	
f ₄	30	[-600, 600] ^a	0	0.1	
f ₅	2	[-100, 100] ^a	0	10 ⁻⁵	

4.2 实验结果及分析

为了使 HDE 算法的测试结果更具对比性,我们将 HDE 与 DE 基本算法,DE 改进算法 DDE 以及 PSO 算法进行了比较.每种算法针对 5 个测试函数都运行 20 次,取其搜索成功率,并计算搜索成功情况下的 nfe 平均值和收敛时间(t).各算法实验结果如表 3~表 7 所示.在表 3~表 7 中,DE7 代表 DE 基本算法 DE/rand/1/bin, DDE7 为基于 DE/rand/1/bin 策略的动态微分进化算法, DDE1 则为基于 DE/best/1/exp 策略的动态微分进化算法. PSO 算法的仿真结果引自参考文献[12],其中 nfe 是通过参考文献[12]所列的迭代次数换算而得到的, nfe = 迭代次数 × (种群规模 + 1),由于文献[12]中未给出 PSO 算法的平均收敛时间,因此此栏空缺.

本文中的所有算法对 5 个测试函数设置的种群数量都为 60 个,最大迭代次数为 10000 次,算法迭代 10000 次还没有达到表 2 所列的停止条件则认为本次搜索失败. DE 算法其他参

数设置为: F = 0.5, CR = 0.5; PSO 算法参数设置如下: w = 0.729; c₁ = c₂ = 1.494.

本文所列 DE 相关算法均采用 C 语言进行编写,开发环境为 LabWindows/ CVI 6.0,这是一个 C 语言编程环境,所用计算机为 IBM T30 笔记本电脑,主频 2.0 GHz,内存 256M,操作系统为 Windows 2000.

表 4 f₂ 函数仿真结果

算法	成功率	nfe	t/s
DE7	100 %	31869	0.5295
DDE7	100 %	32220	0.5108
DDE1	100 %	19860	0.1715
PSO	100 %	36720	—
HDE1	100 %	15437	0.2159
HDE2	100 %	13932	0.2350

表 5 f₃ 函数仿真结果

算法	成功率	nfe	t/s
DE7	100 %	251388	5.6038
DDE7	100 %	237180	5.1002
DDE1	100 %	8580	0.1217
PSO	100 %	10020	—
HDE1	100 %	77770	1.5365
HDE2	100 %	72189	1.5957

表 6 f₄ 函数仿真结果

算法	成功率	nfe	t/s
DE7	100 %	29226	0.7227
DDE7	100 %	26580	0.6385
DDE1	100 %	22680	0.3782
PSO	100 %	17280	—
HDE1	100 %	13591	0.2990
HDE2	100 %	11226	0.2795

表 7 f₅ 函数仿真结果

算法	成功率	nfe	t/s
DE7	100 %	12873	0.0796
DDE7	100 %	12240	0.0835
DDE1	50 %	3120	0.0290
PSO	95 %	19200	—
HDE1	95 %	6392	0.0252
HDE2	90 %	4980	0.0314

从仿真数据可以看出, HDE1 和 HDE2 算法针对 5 个测试函数都取得了很好的效果. HDE 算法在保证搜索成功率的基础上,大大降低了收敛所需目标函数评价次数和收敛时间. HDE 算法对目标函数的评价次数和收敛时间与 DE 基本算法相比减少了 50% 以上. 显而易见, HDE 算法不仅优于 DE 基本算法,总体上也优于 DDE 算法和 PSO 算法. 本文所讨论的 HDE1 算法为 DE7 优化策略和 DE6 优化策略的混合优化, HDE2 算法则将种群中的个体随机的分成两组,一组采用 DE7 优化策略,另一组则采用 DE6 优化策略. 在实际应用中,可以根据实际需要,采用不同的优化策略组合,或采用不同的比例进行混合优化. 例如利用 HDE2 算法针对 f₅ 函数寻优时,如果种群中 80% 个体采用 DE7 优化策略,其余 20% 个体采用 DE6 优化策略,则其搜索成功率可以达到 100%,目标函数的平均评价次数为 7467.

5 结论

本文提出了一种基于混合优化策略的微分进化改进算法—HDE 算法,该算法通过对 DE 算法的不同优化策略进行混合,使算法在保证良好收敛性的同时,提高了搜索效率. 对 5 个测试函数的仿真结果表明 HDE 算法相对于 DE 基本算法、DDE 算法的性能有了较大提高,算法收敛所需目标函数评价次数和收敛时间比 DE 基本算法减少了 50% 以上.

参考文献:

[1] Rainer Storn, Kenneth Price. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces [J]. Journal of Global Optimization, 1997, 11(4): 341 - 359.

- [2] Rainer Storn. Designing nonstandard filters with differential evolution[J]. IEEE Signal Processing Magazine, 2005, 22(1): 103 - 106.
- [3] Chong-wei Chen, De-zhao Chen, Guang-zhi Cao. An improved differential evolution algorithm in training and encoding prior knowledge into feedforward networks with application in chemistry[J]. Chemometrics and Intelligent Laboratory Systems, 2002, 64(1): 27 - 43.
- [4] Sandra Paterlinia, Thiemo Krinkb. Differential evolution and particle swarm optimisation in partitional clustering[J]. Computational Statistics & Data Analysis, 2006, 50(5): 1220 - 1247.
- [5] 冯琦, 周德云. 基于微进化算法的时间最优路径规划[J]. 计算机工程与应用, 2005, 41(12): 74 - 75.
Feng Qi, Zhou De-yun. Time optimal path planning based on differential evolution algorithm[J]. Computer Engineering and Application, 2005, 41(12): 74 - 75. (in Chinese)
- [6] Anyong Qing. Dynamic differential evolution strategy and applications in electromagnetic inverse scattering problems[J]. IEEE Transactions on Geoscience and Remote Sensing, 2006, 44(1): 116 - 125.
- [7] Paul K. Bergey, Cliff Ragsdale. Modified differential evolution: a greedy random strategy for genetic recombination[J]. The International Journal of Management Science, 2005, 33(3): 255 - 265.
- [8] Jianyong Sun, Qingfu Zhang, Edward P K Tsang. DE/EDA: a new evolutionary algorithm for global optimization[J]. Information Sciences, 2005, 169(3 - 4): 249 - 262.
- [9] P Kaelo, M M Ali. A numerical study of some modified differential evolution algorithms[J]. European Journal of Operational Research, 2006, 169(3): 1176 - 1184.
- [10] 李颖, 徐桂芝, 等. 微进化算法在头部电阻抗成像中的应用[J]. 中国生物医学工程学报, 2005, 24(6): 672 - 675.
Li Ying, Xu Gui-zhi, et al. Application of DE algorithm for brain imaging using EIT[J]. Chinese Journal of Biomedical Engineering, 2005, 24(6): 672 - 675. (in Chinese)
- [11] 王凌. 智能优化算法及其应用[M]. 北京:清华大学出版社, 2001. 1 - 6.
- [12] Ioan Cristian Trelea. The particle swarm optimization algorithm: convergence analysis and parameter selection[J]. Information Processing Letters, 2003, 85(6): 317 - 325.

作者简介:



赵光权 男, 1978 年出生于浙江省东阳市, 哈尔滨工业大学自动化测试与控制系在读博士研究生. 主要研究方向: 计算智能、故障诊断. E-mail: hit53zhao@hit.edu.cn



彭喜元 男, 1961 年生于内蒙古四子王旗, 哈尔滨工业大学自动化测试与控制系教授, 博士生导师. 研究方向: 自动测试技术及系统、计算智能、故障诊断技术及应用.