

文章编号: 1001-0920(2004)01-0049-04

差异演化的实验研究

谢晓锋¹, 张文俊¹, 张国瑞², 杨之廉¹

(1. 清华大学 微电子学研究所, 北京 100084; 2. 沈阳化工有限公司, 辽宁 沈阳 110026)

摘要: 首先基于一些实例研究了差异演化(DE)的参数选择问题; 然后在分析DE特点的基础上, 将缩放因子 F 由固定数值设为随机函数, 实现了一个简化的DE版本(SDE)。该方法不仅减少了需调整的参数, 而且对CR的参数选择更为宽松。与已有文献中遗传算法的带约束型数值优化问题的实验结果对比, 表明SDE能在较少的计算次数内获得较好的结果。

关键词: 差异演化; 演化计算; 数值优化**中图分类号:** TP301.6 **文献标识码:** A

Empirical study of differential evolution

XIE Xiaofeng¹, ZHANG Wenjun¹, ZHANG Guorui², YANG Zhilian¹

(1. Institute of Microelectronics, Tsinghua University, Beijing 100084, China; 2. Shenyang Chemical Limited Company, Shenyang 110026, China Correspondent: XIE Xiaofeng, Email: xiexiaofeng@tsinghua.org.cn)

Abstract: The parameters selection of differential evolution (DE) is studied by experiments on some benchmark examples. Then a simplified DE version (SDE) is realized with randomized scaling factor F based on the analysis for the features of DE, which not only reduces a parameter that need to be adjusted, but also is flexible for the selection of parameter CR. The experiments by comparing with genetic algorithm (GA) on some constrained numerical optimization problems show that SDE can get better results in much less evaluation times.

Key words: differential evolution; evolutionary computation; numerical optimization

1 引言

在科学和工程领域中, 经常会遇到连续空间中的数值优化问题¹。它们的目标函数通常是非线性甚至是不可微的, 这时传统的优化方法便很难获得成功。

近 30 年的研究充分表明, 模拟自然进化的搜索过程可产生非常鲁棒的计算机算法, 即使这些模型只是自然界生物体演化的粗糙简化。演化算法(EA)², 如进化规划(EP)³、进化策略(ES)⁴、遗传算法(GA)^{5,6}等, 都是基于这种思想而发展起来的通用的问题求解方法。它在赋予 EA 自组织、自适应、自学习等特征的同时, 并不受搜索空间限制性条

件(如是否可微、连续性等)的约束, 也不需要其他辅助信息(如导数), 不仅能获得较高的效率, 而且具有易于操作和通用的特点。

差异演化(DE)^{7,8}是一种基于群体差异的 EA, 它在首届 IEEE 演化计算竞赛中⁷表现突出, 并获得广泛的应用^{9,10}。

本文首先简要介绍 DE 算法, 然后基于演化计算常用测试函数¹¹对 DE 的参数设置进行研究, 在此基础上提出了简化的 DE(SDE)版本。在加入约束处理规则¹²后, 与文献 1 中 GA 的结果对比, 表明 SDE 能在较少的计算次数内对通常的数值优化问题得到更好的优化性能。

2 DE 算法

DE 最初的群体是随机产生的, 它的每一个个体 x 为 D 维搜索空间中的一个浮点向量 在评估适应值后, 其中最好的个体记为 x_{gbest}

DE 最基本的变异成分是差异向量 每个向量对包括群体中的两个个体 (x_a, x_b) . 差异向量定义为 $D_{ab} = x_a - x_b$

变异过程首先从群体中随机选择 4 个个体 x_a, x_b, x_c, x_d , 然后由这 4 个向量组合成 $D_{abcd} = D_{ab} + D_{cd} = (x_a - x_b) + (x_c - x_d)$. 组合的差异向量 D_{abcd} 反映了群体的差异度 为控制收敛速度, 通常对 D_{abcd} 乘以一个缩放因子 F .

一旦求出 D_{abcd} , 则用它对 x_{gbest} 进行变异操作, 即 $x_\beta = x_{gbest} + F * D_{abcd}$. 这里 x_β 可看作 x_{gbest} 的一个噪音版本 虽然有时将差异向量加到一个随机选择的个体可能更加合适, 但对大多数例子, 使用 x_{gbest} 可以加速收敛 如果使用 x_{gbest} 时导致早熟收敛, 则应考虑使用随机个体

对于群体中第 i 个个体 x_i , 将与 x_β 进行交叉操作产生子代个体 x_T . 为保证个体 x_i 的演化, 首先通过随机选择, 使得 x_T 至少有一位由 x_β 贡献, 而对其他位, 可利用一个交叉因子 CR , 决定 x_T 中哪些位由 x_i 贡献, 哪些位由 x_β 贡献 如果 $RAND(\bullet) < CR$, 则该位由 x_β 贡献, 即 CR 越大, x_β 贡献越多. 当 $CR = 1$ 时, $x_T = x_\beta$

由 x_i 和 x_β 产生的子代 x_T , 将与其父代个体 x_i 进行竞争, 只接受较好的个体, 并替换掉较差的个体, 它的群体规模是不变的

DE 算法的参数包括群体规模 N_c , 缩放因子 F , 交叉因子 CR . 为对参数的作用有直观的了解, 作者对常用的 Rosenbrock 函数和 Griewank 函数¹¹ (最优解均为 0) 进行了测试

Rosenbrock 函数的表达式为

$$f_1(x) = \sum_{d=1}^{D-1} (100(x_{d+1} - x_d^2)^2 + (x_d - 1)^2). \tag{1}$$

Griewank 函数的表达式为

$$f_2(x) = \frac{1}{4000} \sum_{d=1}^D x_d^2 - \prod_{d=1}^D \cos\left(\frac{x_d}{\sqrt{d}}\right) + 1. \tag{2}$$

图 1 和图 2 分别给出了群体规模 $N_c = 20$ 时, Rosenbrock 函数和 Griewank 函数的适应值随 F 和 CR 的变化情况 其中: $F = 0.1 \sim 1, CR = 0 \sim 1$ 函数维数 $D = 20$, 计算代数 $T = 1500$, 适应值取函数值的对数, 对每个参数组合计算 250 次取平均值

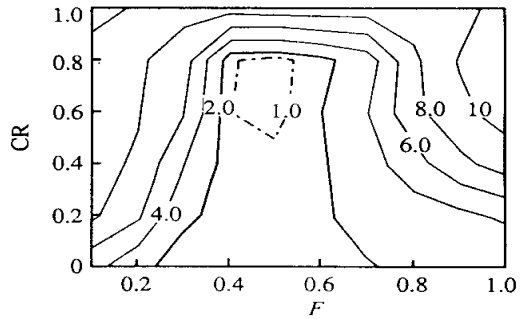


图 1 Rosenbrock 函数的适应值随 F 和 CR 的变化

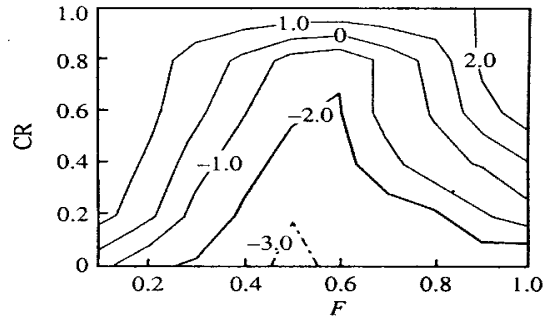


图 2 Griewank 函数的适应值随 F 和 CR 的变化

可以看出, 对于图中的等值线, 数值越小, 适应值越高 算法获得的高适应值 (Rosenbrock 函数小于 1, Griewank 函数小于 - 3), 其参数设置主要位于 $F = 0.5$ 左右, 而 CR 越小, F 的区域越宽 这是因为如果 F 太大, 则群体的差异度不易下降, 使得群体收敛速度变慢; 如果 F 太小, 则群体的差异度过早下降, 使得群体在成熟前收敛 CR 的增加意味着对多变量相关的处理能力加强, 也意味着 x_β 对子代的贡献增多. 即个体间的非线性交互越少, 越不容易形成自组织系统, 因而对复杂问题的演化能力下降¹³. CR 的选取应根据问题的变量相关程度决定 对于不相关的变量, 如 Griewank 函数, 取 0 能达到较好的效果; 对于相关性较大的变量, 如 Rosenbrock 函数, 应取大于 0 的值

3 简化的 DE

与 GA 相比, DE 具有以下特点: 1) 变异基于群体的差异, 由于早期群体差异度通常较大, 使得早期探索能力较强, 而随着代数的增加, 差异度变小, 使得后期开发能力增加, 这类似于一个退火过程; 2) 基于 CR 可控制个体参数的各维对交叉的参与程度, 以及全局与局部搜索能力的平衡; 3) 替换是在父子个体之间以贪婪方式进行的, 类似于排挤模型⁶, 有助于维持群体的多样性

DE 通常采用固定的缩放因子 F , 这涉及到群体差异度减小的速度问题 对于不同的问题需要有不

同的值, 对 F 的调整需要更多的经验. 在 Storn 等⁸ 的测试中, 尽管大部分例子表明取 $F = 0.5$ 能得到好的结果, 但也有少部分需要采用其他值.

为减少用户的参与程度, 需要降低 F 的影响. 最好的解决方案是自适应调整 F . 本文提出一种相对简化的方案, 采用 $F = \text{RAND}(\bullet)$, 即取 $0 \sim 1$ 之间的随机数. 这种方案称为简化的 DE, 即 SDE.

另外, 通常遇到的数值优化问题常带有约束条件, 使用基于惩罚的方法需要引入新的参数, 不易使用. 为此, 本文采用如下约束处理规则¹²: 1) 任何可行点要好于不可行点; 2) 在可行点中优选目标函数更好的点; 3) 在不可行点中优选违反约束小的点.

4 测试与讨论

为研究 DE 在数值优化问题中的应用, 作者对文献 1 中带约束的优化函数实例进行了测试 (只选用其中无等式约束的 8 个实例). 这些测试例子包括不同的目标函数性质 (线性、二次、三次、多项式、非线性等), 以及不同的变量和不同数量的约束. 可行空间占搜索空间的比例 ρ 有很大变化, 可行空间的布局也有很大不同. 这些测试例子列于表 1. 表中给出了变量数目 n , 目标函数性质, 可行区比例 ρ , 不同约束的数量 (其中 LI 为线性不等式, NI 为非线性不等式), 在优化点激活的约束数量 a , 以及各函数的全局优值 F_{opt} .

表 1 测试实例

G	n	性质	$\rho/\%$	LI	NI	a	F_{opt}
G_1	13	quadratic	0.0111	9	0	6	-15.000
G_2	20	nonlinear	unknown	0	2	1	0.80362
G_4	5	quadratic	52.1230	0	6	2	-30.66554
G_6	2	cubic	0.0066	0	2	2	-6.961814
G_7	10	quadratic	0.0003	3	5	6	24.306
G_8	2	nonlinear	0.8560	0	2	0	0.095825
G_9	7	polynomial	0.5121	0	4	2	680.63
G_{10}	8	linear	0.0010	3	3	6	7.04933

以下对 SDE 的测试, 演化代数设为 2 000 代 (其中 G_2 为 10 000 代), 所有结果为 50 次计算取平均.

表 2 和表 3 分别给出了 $N_c = 14, F = 0.5$ 时 DE 和 SDE 的测试结果. 可以看出, 在 CR 为 0 或 0.1 时, DE 和 SDE 得到相近的结果. DE 在 CR = 1 时几乎失去优化能力, 而 SDE 还有较强的优化能力. 这是因为随机的 F 增加了算法的遍历性, 从而使 SDE 的参数选择更为宽松. 相比之下, 算法在 CR 为一个小的数值时 (如 0.1), 对大多数情形能获得最好的效

果.

表 4 给出了 $N_c = 70$, SDE 在 CR = 0.1 时的测试结果, 并与文献 1 中实验 1 的 GA 结果进行比较. 对于 GA, 种群大小 $N_{\text{pop}} = 70$, 代沟 $\text{gap} = 100\%$, $T = 5\,000$ (G_2 为 10 000), 总评估次数 $T_e = N_{\text{pop}} * \text{gap} * T = 350\,000$ (G_2 为 700 000); 对于 DE, $T_e = N_c * T = 140\,000$ (G_2 为 700 000). 可以看出, 在较少的计算次数内, SDE 对所有函数均能取得比 GA 更好的结果.

表 2 $N_c = 14$ 时 DE 对不同 CR 的平均测试结果

G	CR = 0	CR = 0.1	CR = 1
G_1	-14.9724	-15.000	-5.3047
G_2	0.62589	0.76531	0.22373
G_4	-29.951309	-30.652870	-30.199138
G_6	-6.673702	-6.961814	-6.727506
G_7	32.914	25.753	92% failed
G_8	0.09449	0.09182	0.08482
G_9	687.84	682.28	6.89951
G_{10}	7.4249	7.6285	82% failed

表 3 $N_c = 14$ 时 SDE 对不同 CR 的平均测试结果

G	CR = 0	CR = 0.1	CR = 1
G_1	-15.000	-14.9996	-13.7263
G_2	0.61217	0.77694	0.40746
G_4	-30.042597	-30.650260	-30.561152
G_6	-6.729893	-6.961717	-6.958807
G_7	30.927	25.635	27.306
G_8	0.09316	0.08902	0.07698
G_9	688.76	682.22	681.06
G_{10}	7.4804	7.8161	8.7263

表 4 $N_c = 70$ 时 SDE 与 GA 平均测试结果比较

G	GA ¹¹	SDE (CR = 0.1)
G_1	-14.4609	-14.9982
G_2	0.79176	0.79605
G_4	-30.6438	-30.644273
G_6	-6.1912	-6.960591
G_7	26.619	25.23893
G_8	0.087155	0.095825
G_9	682.18	681.12
G_{10}	9.1417	7.6653

图 3~ 图 6 给出了在不同 N_c 和 CR 下, SDE 对较难的几个问题 G_2, G_7, G_9 和 G_{10} 的相对适应值 F_{norm} 随演化代数的变化情况. 这里 $F_{\text{norm}} = |F_{\text{best}} -$

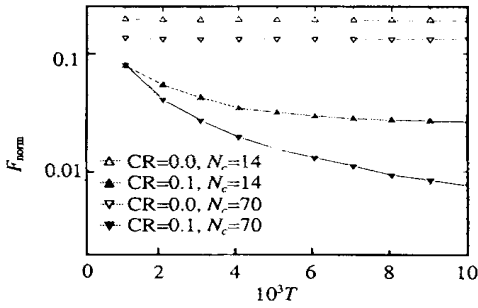


图3 不同Nc和CR时SDE对G2的优化性能

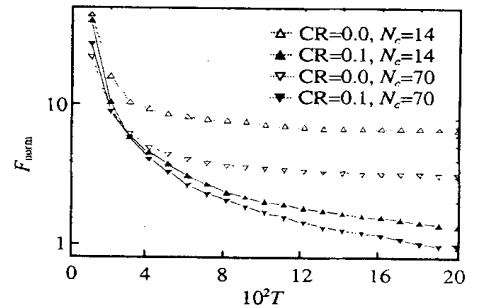


图4 不同Nc和CR时SDE对G7的优化性能

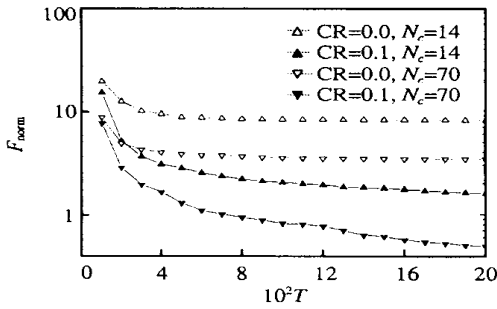


图5 不同Nc和CR时SDE对G9的优化性能

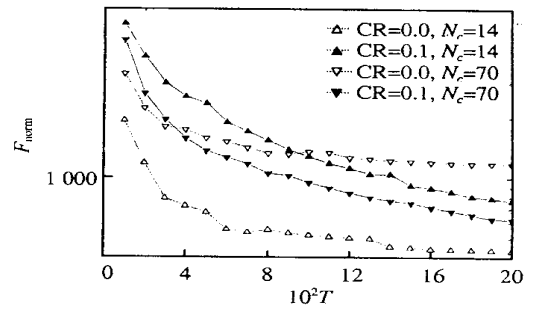


图6 不同Nc和CR时SDE对G10的优化性能

F_{opt} ，其中 F_{best} 为到当前演化代为止的最好适应值

可以看出,当 N_c 较大时,总体结果较好,但对部分函数(如 G_{10})得到的结果反而较差 这是由于群体较大时,群体差异度不易缩小 虽然这使得算法的全局性增加,但也使收敛速度下降 另外,对于大部分情形,CR 的影响比 N_c 更大

5 结 论

本文首先对具体实例进行测试,研究DE 算法中最主要的参数CR 和 F ; 然后通过分析DE 的特点,提出了 $F = RAND(\cdot)$ 的SDE 版本 该方法不仅减少了所需调整的参数,而且对CR 的选择更为宽松 在加入约束处理规则后,与文献1 中GA 带约束型数值优化实例的实验结果进行对比,表明SDE 能在较短的计算时间内获得较好的结果

参考文献(References):

- 1 Koziel S, Michalewicz Z. Evolutionary algorithms, homomorphous mappings and constrained parameter optimization [J]. *Evolutionary Computation*, 1999, 7(1): 19-44
- 2 Whitley D. An overview of evolutionary algorithms: Practical issues and common pitfalls [J]. *Information and Software Technology*, 2001, 43(14): 817-831.
- 3 Fogel L J, Owens A J, Walsh M J. *Artificial Intelligence Through Simulated Evolution* [M].

Chichester: John Wiley, 1996

- 4 Rechenberg I. *Evolutionstrategie: Optimierung Technischer Systems nach Prinzipien der Biologischen Evolution* [M]. Stuttgart: Frommann-Holzboog Verlag, 1973
- 5 Holland J H. *Adaptation in Natural and Artificial Systems* [M]. Ann Arbor: University of Michigan Press, 1975
- 6 De Jong K A. The analysis of the behavior of a class of genetic adaptive systems [D]. Ann Arbor: University of Michigan, 1975
- 7 Storn R. Differential evolution design of an IIR-filter [A]. *IEEE Int Conf on Evolutionary Computation* [C]. Nagoya, 1996: 268-273
- 8 Storn R, Price K. Differential evolution — A simple and efficient heuristic for global optimization over continuous spaces [J]. *J of Global Optimization*, 1997, 11(4): 341-359.
- 9 Pahner U, Hameyer K. Adaptive coupling of differential evolution and multiquadrics approximation for the tuning of the optimization process [J]. *IEEE Trans on Magnetics*, 2000, 36(4): 1047-1051.
- 10 Cheng S L, Hwang C. Optimal approximation of linear systems by a differential evolution algorithm [J]. *IEEE Trans on Systems, Man and Cybernetics — Part A*, 2001, 31(6): 698-707.

(上接第 52 页)

- 11 Shi Y H, Eberhart R C. Fuzzy adaptive particle swarm optimization [A]. *IEEE Int Conf on Evolutionary Computation* [C]. Seoul, 2001. 101-106
- 12 Deb K. An efficient constraint handling method for genetic algorithms [J]. *Comput Methods in Applied Mechanics and Engineering*, 2000, 186 (2-4): 311-338
- 13 Nicolis G, Prigogine I. *Self-organization in Nonequilibrium Systems: From Dissipative Systems to Order through Fluctuations* [M]. New York: John Wiley, 1977.