Journal of Computer Applications
计算机应用,2015,35(5):1333 - 1335
ISSN 1001-9081
CODEN JYIIDU
2015-05-10
http://www.joca.cn

文章编号:1001-9081(2015)05-1333-03
doi:10.11772/j.issn.1001-9081.2015.05.1333

朴素差分进化算法

汪慎文[1,2*],张文生[2],秦 进[3],谢承旺[4],郭肇禄[5]

(1.石家庄经济学院 信息工程学院,石家庄 050031; 2.中国科学院 自动化研究所,北京 100190;
3.贵州大学 计算机学院,贵阳 550025; 4.华东交通大学 软件学院,南昌 330013;
5.江西理工大学 理学院,江西 赣州 341000)

(*通信作者电子邮箱 wangshenwen@whu.edu.cn)

摘 要:针对变异算子学习方式的单一性,提出一种朴素变异算子,其基本思想是向优秀的个体靠近,同时远离较差个体,其实现方式是设计一种缩放因子调整策略,如果三个随机个体在某维上比较接近,则缩放因子变小,反之变大。在实验过程中通过平均适应度评价次数、成功运行次数和加速比等指标表明,基于朴素变异算子的差分进化算法能有效提高算法的收敛速度和健壮性。

关键词:差分进化;朴素变异算子;缩放因子;集成进化

中图分类号:TP301 **文献标志码**:A

Naïve differential evolution algorithm

WANG Shenwen[1,2*], ZHANG Wensheng[2], QIN Jin[3], XIE Chengwang[4], GUO Zhaolu[5]

(1. *School of Information Engineering, Shijiazhuang University of Economics, Shijiazhuang Hebei* 050031, *China*;
2. *Institute of Automation, Chinese Academy of Sciences, Beijing* 100190, *China*;
3. *School of Computer, Guizhou University, Guiyang Guizhou* 550025, *China*;
4. *School of Software, East China Jiaotong University, Nanchang Jiangxi* 330013, *China*;
5. *School of Science, Jiangxi University of Science and Technology, Ganzhou Guangdong* 34100, *China*)

Abstract: In order to solve singleness of mutation study, a naïve mutation strategy was proposed to approach the best individual and depart the worst one. So, a scale factor self-adaptation mechanism was used and the parameter was set to a small value when the dimension value of three random individuals is very close to each other, otherwise, set it to a large value. The results showed that the Differential Evolution (DE) with the new mechanism exhibits a robust convergence behavior measured by average number of fitness evaluations, successful running rate and acceleration rate.

Key words: Differential Evolution (DE); naïve mutation operator; scale factor; integrated evolution

0 引言

差分进化(Differential Evolution,DE)算法[1],是由 Storn 和 Price 提出的一种新型进化算法,因其算法简单易执行、性能卓越等优点引起众多研究者的广泛关注。目前研究者从三个角度对差分进化算法进行改进[2]:1)操作算子的改进,提出一些新的参数控制方法(如:模糊适应差分进化算法(Fuzzy Adaptive DE, FADE)[3]、JADE(Adaptive DE with optional External Archive)[4]、自适应 DE(Self-Adaptive DE, jDE)[5]等)和新设计的差分变异策略(如:三角变异策略[6]等);2)基于静态知识指导的差分进化集成算法,该类算法主要是指 DE 与其他具有某种优良特性的策略集成,如:与分布评估算法(Estimation of Distribution Algorithm, EDA)集成的 DE/EDA[7]、与反向学习策略集成的 ODE(Opposition-based DE)[8]等;3)基于动态知识指导的差分进化集成算法,该类算法对进化过程中的不同变异策略产生的新解评估,其结果用于指导下一代策略选择,如:EPSDE(Ensemble of Mutation Strategies and Control Parameters with DE)[9]、SaDE(Self-adaptive DE)[10]等。

差分进化算法在执行的初期,由于种群个体的差异性较大,变异算子使得算法具有较强的全局搜索能力,在算法执行的后期,个体差异性小,使得算法具有较强的局部搜索能力。这种向种群个体学习的能力使得差分进化算法拥有其他进化算法不可比拟的优点。但是这种学习方式过于单一,不论随机邻居个体是优秀个体还是不好个体,向随机邻居个体学习的方式均相同。于是有研究者提出向全局最优个体[2]或者精英个体学习[11]。这种改进利用群体中的较好个体指导其他个体进化是合理的,但是放弃了不好个体的关于搜索空间的信息,与自然界中的生物利用信息的方式有所不同,因此可能也是低效的。实际上每个个体可以利用所有个体的提供关

收稿日期:2014-12-29;**修回日期**:2015-02-06。

基金项目:国家自然科学基金资助项目(61165004,61402481);河北省青年拔尖人才支持计划项目(冀字[2013]);河北省自然科学基金资助项目(F2015403046);河北省科技支撑计划项目(13210331);河北省教育厅青年科学基金资助项目(QN20131053);石家庄经济学院博士科研启动基金资助项目(BQ201322);江西省教育厅青年科学基金资助项目(GJJ14456,GJJ14373)。

作者简介:汪慎文(1979 -),男,湖北红安人,副教授,博士,CCF 会员,主要研究方向:智能计算、机器学习; 张文生(1966 -),男,河南郑州人,研究员,博士生导师,博士,主要研究方向:模式识别、机器学习; 秦进(1978 -),男,贵州黔西人,副教授,博士,主要研究方向:智能计算; 谢承旺(1974 -),男,湖北武汉人,副教授,博士,主要研究方向:智能计算; 郭肇禄(1984 -),男,江西南康人,讲师,博士,主要研究方向:智能计算、并行计算。

于搜索空间的信息,甚至是来自一个更差的个体的信息。受Qin等[12]研究成果的启发,本文尝试建立一种变异新模式,称之为朴素变异算子(naïve mutation operator):向优秀的个体学习,换句话说,向更优秀的个体靠近;同时排斥更差的个体,换句话说,远离更差的个体。这意味着个体通过和其他随机个体的交流来调整自己的位置。这样一来,每个个体与其他个体的交互显得更自然、更简单、更朴素。

1 基于朴素变异算子的差分进化算法

Qin等[12]利用多Agent系统作为粒子群体行为的建模和仿真工具,研究粒子群体行为的演化动力学。理论和实践都证明,Agent利用靠近和远离两种基本动作在全局勘探和局部开采中找到一个动态平衡,从而找到全局最优位置。

尽管文献[12]以粒子群的群体作为研究对象,但是该结论也适用于基于种群的差分进化算法。在传统的差分进化算法中,每个个体靠近它的随机个体;而在朴素差分进化算法中,学习的方式有两种:靠近好的随机个体和远离差的随机个体。

1.1 朴素变异算子

以"DE/rand/1"算子为基础,给出朴素变异策略"naïve DE/rand/1"的描述。如果3个随机个体在某维上比较接近,则选择较小的缩放因子,使得变异个体在该维上靠近3个个体,如果随机3个体在某维上相差较大,则选择较大的缩放因子,使得变异个体在该维上远离,其数学公式表示如下:

$$v_{i,j}(t) = x_{r1,j}(t) + f_j(x_{r2,j}(t) - x_{r3,j}(t)) \tag{1}$$

其中:$r1,r2,r3$是从集合$\{1,2,\cdots,NP\}\setminus\{i\}$中随机选择的相互不等的整数,$i \in \{1,2,\cdots,NP\}$,$t$为迭代的代数,$j \in \{1,2,\cdots,D\}$,缩放因子$F$的定义如下:

$$\begin{aligned} a_j(t) &= \min(x_{1,j}(t),x_{2,j}(t),\cdots,x_{NP,j}(t)) \\ b_j(t) &= \max(x_{1,j}(t),x_{2,j}(t),\cdots,x_{NP,j}(t)) \\ c_j(t) &= \min(x_{r1,j}(t),x_{r2,j}(t),\cdots,x_{r3,j}(t)) \\ d_j(t) &= \max(x_{r1,j}(t),x_{r2,j}(t),\cdots,x_{r3,j}(t)) \\ f_j(t) &= \frac{d_j(t) - c_j(t)}{b_j(t) - a_j(t)} \end{aligned} \tag{2}$$

1.2 朴素差分进化算法

借鉴传统的DE算法框架,朴素差分进化算法(nDE)的伪代码如下。

算法1 朴素差分进化算法(nDE)。

1)设置种群规模NP、维数D、杂交概率Cr、最大适应度评价次数$MaxFES$、精度VTR。

2)使用公式$x_{ij}(0) = L_j + rand_{ij}[0,1](U_j - L_j)$初始化种群个体并评估个体,令$FES = NP$。

3)变异:对个体用式(1)、(2)进行突变。

4)修补:对越界个体用如下公式处理:

$$v_{ij} = \begin{cases} \min\{U_j, 2L_j - v_{ij}(t)\}, & v_{ij}(t) < L_j \\ \max\{L_j, 2U_j - v_{ij}(t)\}, & v_{ij}(t) > U_j \end{cases}$$

5)杂交:对种群个体用如下公式进行杂交:

$$u_{ij}(t) = \begin{cases} v_{ij}(t), & rand_j(0,1) < Cr \text{ or } j = j_{rand} \\ x_{ij}(t), & \text{其他} \end{cases}$$

7)选择:对个体用如下公式一一竞争选择:

$$X_i(t+1) = \begin{cases} X_i(t), & f(X_i(t)) \leqslant f(U_i(t)) \\ U_i(t), & f(U_i(t)) < f(X_i(t)) \end{cases}$$

8)当适应度评价次数小于最大适应度评价次数或者最好个体适应度值精度大于预定义精度,令$FES = FES + NP$,转3)。

2 实验研究

本文选择其中13个有代表性的经典基准函数来测试算法的有效性,关于13个Benchmark函数的详细描述,参考文献[13]。

所有的实验均在操作系统为Windows 7(64 b),双核2.40 GHz的Intel处理器和2 GB内存,Matlab R2013a的平台上完成。实验中各算法所用到的参数均相同,设置见表1和表2。

表1 实验参数设置

参数	值	参数	值
种群规模	30	运行次数	50
维数	30	算子	naïve DE/rand/1
杂交概率	0.2	缩放因子F	self-adaptation

表2 预定义精度VTR和最大适应度评价次数MaxFEs

函数	*MaxFES*	*VTR*	函数	*MaxFES*	*VTR*
f_1	2E5	1E-50	f_8	3E5	1E-8
f_2	2E5	1E-50	f_9	3E5	1E-10
f_3	5E5	1E-8	f_{10}	1.5E5	1E-10
f_4	5E5	1E-10	f_{11}	1.5E5	1E-20
f_5	5E5	1E-8	f_{12}	1.5E5	1E-8
f_6	1.5E5	1E-8	f_{13}	1.5E5	1E-8
f_7	3E5	1E-3			

为了测试算法设计的有效性,本文选择几种模型进行比较,分别为:传统的应用"DE/rand/1"变异策略的DE算法、应用"naïve DE/rand/1"变异策略的nDE算法、以及来自进化计算领域内研究人员广泛认可的jDE[5]和SaDE[10]。本文使用适应度评价次数(Fitness Evaluation,FE)、成功运行次数(Successful Running Rate,SR)和加速比(Acceleration Rate,AR)三个指标考察算法的收敛速度和算法的健壮性。

2.1 4种算法收敛速度的比较

表3给出了DE、jDE、SaDE、nDE在13个测试函数50次独立运行后得到的实验统计结果。其中,"Mean"和"Std Dev"分别表示算法终止时适应度评价次数的平均值和标准差,"NA"表示算法在MaxFEs内均未成功收敛到预定义的精度,"AVER"表示50次独立运行,算法终止时所有函数上耗费的适应度评价次数的平均值。

从表3的"AVER"值可以看到,在相同的情况下,jDE所需要的代价最小,只需要14.8万次;nDE居其次,需要14.9万次,说明nDE算法总体性能好于DE和SaDE,差于jDE。再从表3中的细节数据也看出,DE共在5个函数50次的运行都不能收敛,nDE共在3个函数50次的运行都不能收敛,SaDE在2个函数50次运行都不能成功收敛,jDE在1个函数50次运行都不能成功收敛。从表3中可以看出,nDE在多模函数上(f_8,f_9,f_{10},f_{11},f_{12},f_{13})都好于其他3个算法,这说明在面对形如起伏的山岭的优化问题时,缩放因子能根据优化问题当前的状态,合理调整参数。

表 3　nDE 与 DE、jDE、SaDE 的适应度评价次数统计结果

函数	DE	jDE	SaDE	nDE
	Mean ± Std Dev	Mean ± Std Dev	Mean ± Std Dev	Mean ± Std Dev
f_1	NA	7.98E+04 ±1.44E+03	9.80E+04 ±3.05E+03	1.02E+05 ±8.43E+02
f_2	NA	1.27E+05 ±1.86E+03	1.38E+05 ±5.15E+03	1.53E+05 ±1.08E+03
f_3	NA	2.10E+05 ±1.49E+04	3.69E+05 ±3.27E+04	NA
f_4	4.53E+05 ±6.43E+03	NA	4.21E+05 ±8.13E+04	1.54E+05 ±2.59E+03
f_5	NA	2.82E+05 ±1.08E+05	NA	NA
f_6	1.92E+04 ±6.59E+02	2.69E+04 ±4.78E+04	1.04E+04 ±4.12E+03	8.47E+03 ±3.46E+02
f_7	NA	2.93E+05 ±2.56E+04	NA	NA
f_8	6.44E+04 ±3.40E+04	1.24E+05 ±1.34E+05	8.93E+04 ±1.06E+05	5.07E+04 ±7.43E+04
f_9	1.79E+05 ±3.26E+04	6.46E+04 ±7.93E+04	2.00E+05 ±1.24E+05	1.46E+05 ±1.12E+05
f_{10}	9.70E+05 ±1.16E+03	3.46E+04 ±8.63E+02	1.27E+05 ±4.47E+04	4.35E+04 ±5.25E+02
f_{11}	8.98E+04 ±2.60E+03	4.92E+04 ±4.45E+04	9.53E+04 ±5.75E+04	3.91E+04 ±1.39E+03
f_{12}	5.12E+04 ±1.27E+03	3.35E+04 ±4.35E+04	6.78E+04 ±6.24E+04	2.13E+04 ±5.22E+02
f_{13}	6.37E+04 ±1.54E+03	1.01E+05 ±6.36E+04	1.28E+05 ±4.77E+04	2.85E+04 ±1.76E+04
AVER	2.09E+05 ±6.18E+0.3	1.48E+05 ±4.34E+04	1.96E+05 ±4.37E+04	1.57E+05 ±1.62E+04

2.2　四种算法成功次数和加速比比较

表 4 给出了 nDE 和 DE、jDE、SaDE 在测试函数成功收敛的运行次数 SR 和加速比 AR 统计结果,其中,“(4)v(1)”表示 nDE 算法与 DE 算法比较,以此类推。“NA”表示在相比中,被比较算法在 50 次独立运行中均未收敛到预定义的精度(VTR)。

表 4　nDE 与 DE、jDE、SaDE 成功运行次数和加速比统计结果

函数	SR				AR		
	DE	jDE	SaDE	nDE	(4)v(1)	(4)v(2)	(4)v(3)
f_1	0	50	50	50	NA	1.00	1.00
f_2	0	50	50	50	NA	1.00	1.00
f_3	0	50	50	0	NA	0.00	0.00
f_4	50	0	32	50	1.00	NA	1.56
f_5	0	42	0	0	NA	0.00	NA
f_6	50	44	50	50	1.00	1.14	1.00
f_7	0	5	0	0	NA	0.00	NA
f_8	49	32	40	46	0.94	1.44	1.15
f_9	47	45	20	33	0.70	0.73	1.65
f_{10}	50	50	11	50	1.00	1.00	4.55
f_{11}	50	42	24	50	1.00	1.19	2.08
f_{12}	50	44	32	50	1.00	1.14	1.56
f_{13}	50	19	9	49	0.98	2.58	5.44
AVER	30.46	36.38	28.31	36.77			

从表 4 的“AVER”值可以看出,nDE 成功运行次数的平均值为 36.77,高于 DE 的 30.46、jDE 的 36.38 和 SaDE 的 28.31,这说明 nDE 具有较强的健壮性。nDE 在能成功收敛的函数上,成功收敛的次数都很高,只有在一个函数上是 33 次,其他值均在 44 次以上,更进一步说明该策略的有效性。

3　结语

针对差分进化算法的学习方式过于单一,不论随机邻居个体是优秀个体还是不好个体,向随机邻居个体学习的方式均相同,以靠近好的随机邻居、远离差的随机邻居为出发点,本文提出一种新的缩放因子确定机制,设置比较小的缩放因子进行靠近,比较大的缩放因子予以远离,并应用于差分进化算法中。通过在 13 个测试函数上的运算比较,朴素差分进化算法具有更高的收敛速度和鲁棒性。

参考文献:

[1] STORN R, PRICE K. Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces, TR-95-012[R]. Berkeley: International Computer Science Institute, 1995.

[2] WANG S, DING L, ZHANG W, *et al.* Survey of differential evolution[J]. Journal of Wuhan University: Natural Science Edition, 2014, 60(4): 283 - 292(汪慎文, 丁立新, 张文生, 等. 差分进化算法研究进展[J]. 武汉大学学报: 理学版, 2014, 60(4): 283 - 292.)

[3] LIU J. A fuzzy adaptive differential evolution algorithm[J]. Soft Computing, 2005, 9(6): 448 - 462.

[4] ZHANG J., SANDERSON A. JADE: adaptive differential evolution with optional external archive[J]. IEEE Transactions on Evolutionary Computation, 2009, 13(5): 945 - 958.

[5] 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.

[6] FAN L J, LAMPINEN J. A trigonometric mutation operator to differential evolution[J]. Journal of Global optimization, 2003, 27(1): 105 - 129.

[7] SUN J, ZHANG Q, TSANG E. DE/EDA: a new evolutionary algorithm for global optimization[J]. Information Sciences, 2005, 169(3): 249 - 262.

[8] RAHNAMAYAN S, TIZHOOSH H, SALAMA M. Opposition-based differential evolution[J]. IEEE Transactions on Evolutionary Computation, 2008, 12(1): 64 - 79.

[9] MALLIPEDDI R, SUGANTHAN P, PAN Q, *et al.* Differential evolution algorithm with ensemble of parameters and mutation strategies [J]. Applied Soft Computing, 2011, 11(2): 1679 - 1696.

[10] QIN A, HUANG V, SUGANTHAN P. Differential evolution algorithm with strategy adaptation for global numerical optimization[J]. IEEE Transactions on Evolutionary Computation, 2009, 13(2): 398 - 417.

[11] WANG S, DUAN Y, SHU W, *et al.* Differential evolution with elite mutation strategy[J]. Journal of Computational Information Systems, 2013, 9(3): 855 - 862.

[12] QIN J, LIANG Z. A naïve particle swarm optimization[C]// Proceedings of the 2012 IEEE Congress on Evolutionary Computation. Piscataway: IEEE, 2012: 163 - 170.

[13] YAO X, LIU Y, LIU G. Evolutionary programming made faster [J]. IEEE Transactions on Evolutionary Computation, 1999, 3(2): 82 - 102.