

一种新的求解多峰函数优化问题的动态演化算法^{*})

覃俊^{1,2} 康立山² 陈毓屏²(中南民族大学计算机学院 武汉430074)¹ (武汉大学软件工程国家重点实验室 武汉430072)²

摘要 本文给出了一种新的求解多峰函数优化问题的定义:定位所有的极值点,包括全局的峰值点和局部的峰值点。传统的演化算法框架都是群体固定的演化迭代过程,对求解多峰函数优化问题时由于无法事先得知峰值点的个数而很难确定合适的群体大小,影响了算法的效率。提出一种群体动态可调的演化方式,使得初始群体大小可任意指定,在演化过程中通过聚集和按比例引入新个体两个过程而动态变化。实验表明,该算法能尽可能多地定位峰值点。

关键词 动态群体,演化算法,多峰函数

A Novel Dynamic-Population Based Evolutionary Algorithm for Multimodal Function Optimization

QIN Jun^{1,2} KANG Li-Shan¹ CHEN Yu-Ping¹(Computer College, South China University For Nationalities, Wuhan 430074)¹(The State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072)²

Abstract The traditional evolutionary algorithm with a fixed-size population is not suitable especially for solving multimodal function optimization because it's impossible to know the number of solution in advance and hence it's difficult to specify a suitable size of population. In this paper, a novel algorithm with dynamic population is presented. In the process of evolution, the size of population is tuned by a aggregation and introduction of new individuals. A initial experiment is given.

Keywords Dynamic population, Evolutionary algorithm, Multimodal function

1 引言

当前,多峰函数优化问题由于其广泛的应用技术背景一直引起了优化界的注意。多峰函数一般是指有多个峰值点的函数^[1],这些峰值点的函数值可能是相同的,也可能不同。当前研究多峰函数时主要考虑的是搜寻有相同函数值的多个峰值点(全局峰值点)。另一种同样重要但未被引起足够重视的研究内容是搜寻所有的峰值点而不论它们是否是全局极值点。这在实际应用中也是有广泛的应用背景,如由于各种客观原因无法使用全局极值点定位的最优解决方案时,局部峰值点也可以作为候选解。所以以一种较为可行的办法搜寻到所有的峰值点(包括局部峰值点)是本文要考虑的主要问题。

演化算法是一种受达尔文生物演化理论启发的随机计算模型,通过反复迭代,那些对环境适应能力强的个体生存下来。该模型作为优化方法的主要特点是不需要其他如导数等预知信息,仅仅只需知道环境信息(对函数优化来说是目标函数值),所以特别适合对一些非线性和超高维问题的优化。但是,演化算法在设计实现时有一个主要的难点,那就是各种参数的选取(如:群体规模、停机条件、遗传操作概率等)没有一个通用的办法来判断和确定,这就阻碍了演化算法的实用性。

当考虑用演化算法定位多峰函数的所有峰值点时,该问题显得尤为重要。由于群体规模直接对应待寻找的峰值点,所以如果群体规模选取不当则直接影响算法性能。群体规模过小则无法找到所有的峰值点,过大则又浪费了计算资源,影响算法的效率。

根据以上分析,我们提出一种群体规模动态可调的演化框架,群体规模初始时可任意选取,在演化过程中被一个称为“聚集”和“按比例的新个体引入”的过程控制,既能全面搜索找到尽可能多的峰值点,又保证没有多余的资源被浪费(体现在每个山峰只有一个个体定位)。

2 多峰函数求解的相关工作

一个典型的多峰函数的例子(不失一般性,本文总以函数极大化为例)如,例1:

$g_1(x) = \sin^6(5\pi x)$, $x \in [0, 1]$ 。该函数的适应值图如图1所示。该例子在定义域内有多个峰值点,且每个峰值点都是全局峰值点(它们的函数值相同)。又如,例2:

$g_2 = 2^{(-2((x-0.1)/0.9)^2)} \sin^6(5\pi x)$, $x \in [0, 1]$ 。该函数的适应值图如图2,在定义域内也有多个峰值点,但除了在点 $X = (0.9, 9)$ 处是全局峰值点外,其它的山峰都是局部峰值点。我们希望能定位所有这些峰值点。

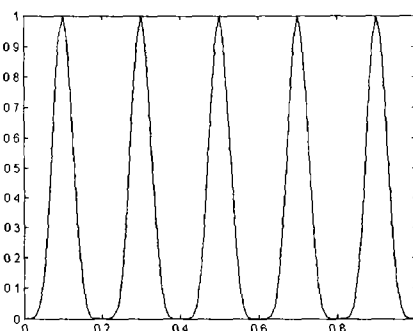


图1

多峰函数优化大多考虑的是寻找尽可能多的全局峰值点(如果有多个的话),一般没有考虑局部峰值点。如文[2]利用距离的概念保留每一代的最好的且相隔有一定距离的为“种子”集合,类似于精英个体(elitist)集合^[3],然后对上述“种子”进行常规的遗传操作,反复演化;文[4]提出了基于区域收缩的新思想,使得经过优化后的个体在演化后期只在各自局部

^{*}) 基金项目:国家自然科学基金(69635030, 60073043, 70071042)。覃俊 副教授,博士生,主要研究领域:遗传算法、电子商务、数据库与KDD。

寻找峰值点。文[1]利用生物免疫的观点保持了群体多样性,这些研究工作都是针对具有多个峰值点但峰值一样(即峰值点都是全局极值点)的函数而设计,没有考虑局部峰值点的情况;并且,群体大小相对固定,无法找到比初始群体规模更多的峰值点(如果有的话)。

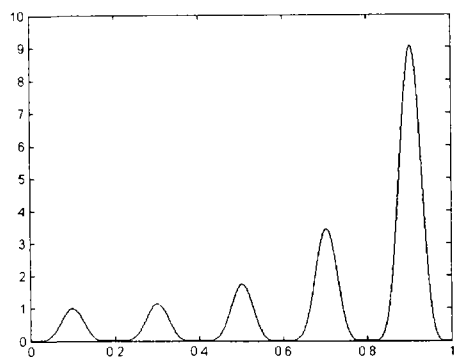


图2

在这些文献中考虑得最多的就是如何保持群体的多样性,来保证全局搜索性能,以便定位所有的全局峰值点。但是这些算法的效果明显依赖于设计者对全局峰值点个数的预测。如果事先对峰值点个数完全不清楚,则算法的效果不能得到保证。

3 算法基本思想

本文针对上述情况,提出了本文关心的两个问题:(1)希望定位所有的峰值点(包括局部峰值点);(2)希望在对待优化问题的峰值点个数不清楚的情况下,算法能动态调整群体大小,使群体大小能最终收敛到与峰值点个数相当的规模。下面给出本文的定义。

定义1 多峰函数优化问题的任务是寻找函数所有的峰值点,包括全局极值点和局部极值点。

在预先不清楚峰值点个数的情况下,固定群体规模是不合适的。我们希望能演化过程中利用环境信息自适应调整群体大小。基本思想如下:任意设定群体规模,对当前群体中的每一个个体进行局部搜索(扰动),希望当前个体能爬到它附近的邻域峰值点(当然有可能也是全局峰值点),因为群体规模任意给定,所以群体规模可能过大,体现在多个个体聚集在单个峰值点上,造成拥挤现象^[3];也可能群体规模过小,还有其它的峰值点没有搜索到。为了避免这两种情况,我们利用“聚集”来避免前一种情况,即基于某种距离概念(本文使用欧式空间)保证个体之间不太拥挤;为避免后一种情况,我们利用随机引入新个体来加强全局搜索能力。更重要的是,我们引入的新个体的数目多少的确定并不是为了补充被抑制掉的个体,好保证群体规模在每次迭代时相同,而这种方法是当前处理类似问题的通用方法;由于我们的目的是要尽可能多的找到峰值点,且不希望找到的峰值点个数局限于预先固定的群体规模,我们提出按比例产生新个体的想法,即

$$\text{PopSize}(i) = \text{Cluster}(i) + \epsilon * \text{Cluster}(i) \quad (1)$$

其中,PopSize(i)表示第i代的群体规模,Cluster(i)表示在第i代进行聚集操作后所剩个体数,右边第二项表示新引入的个体数目。 ϵ 是一个比例因子, $\epsilon \in [0, 1]$ 。这样一来,群体规模在每次迭代时是变化的,因此,动态变化的群体规模可以自适应地跟踪当前已定位到的峰值点个数。

为了保持演化策略的局部微调和全局搜索之间的动态平衡,以保证算法的稳定性,我们借鉴免疫生物学中的浓度饱和现象,只有当前群体“浓度”饱和(平均适应值在相邻的两代相差不大)的情况下,聚集和新个体的引入才被允许。换句话说,

只有当前群体浓度饱和了,我们假定当前个体已爬到了各自所在附近的山峰上了。

我们称当前群体(设第i代)的浓度 D_i 已饱和,如果:

$$|D_i - D_{i-1}| < d, \quad (2)$$

其中群体浓度定义为群体的平均适应值:

$$D_i = \frac{\sum_{j=1}^{\text{PopSize}(i)} f^*(j)}{\text{PopSize}(i)} \quad (3)$$

$f^*(j)$ 表示个体j规格化后的适应值(目标函数值), $f^*(j) \in [0, 1]$, d 是饱和参数,若新旧群体的浓度变化小于此值,表明该群体浓度已经饱和,无进一步微调的必要,需要利用聚集操作进行抑制,去掉拥挤的差个体,然后引入新的个体(大小与聚集后的群体大小成比例)若当前群体的浓度变化大于 d ,则表明当前群体的浓度还未饱和,还有发展空间,个体还需要继续爬山以达到附近峰值点。

值得说明的是,新个体的引入不一定会造成群体规模的单调递增,因为每次迭代都有一个丢弃部分拥挤个体的聚集过程。

3.1 算法基本框架

由于规模受浓度抑制和按比例引入新个体两种因素影响,群体规模在演化中自适应地调整,算法效果与初始群体规模大小关系不大。算法基本框架如下:

第一步,随机产生初始群体,个数随意;

第二步,当(不满足终止条件时)做

2.1 让每个个体在邻域进行扰动;

2.2 计算当前群体的浓度;

2.3 如果群体浓度还未饱和,则转2.1,继续局部爬山过程;否则,

2.4 对当前群体进行聚集,去掉多余个体;

2.5 对剩下的个体按比例随机产生新个体;

第三步,结束

3.2 算法有关说明

在2.1步,算法相当于在进行局部微调,在2.4和2.5步,算法进行多样性保持和全局搜索工作。

个体的局部扰动策略可以有多种,在这里相当于演化策略中的变异算子,我们采用的是对文[5]中提出的标准变异算子加以改进。基本思想是:与较好的个体相比,让较差的个体扰动的幅度也较大。

$$x' = x + \exp(-\rho * f^*(x)) * N(0, 1) \quad (4)$$

其中, x' 是对个体 x 扰动后得到的, $N(0, 1)$ 是均值为零,方差为 $\sigma=1$ 的高斯随机变量。 ρ 控制该指数函数的衰减,该指数函数代表扰动幅度。 $f^*(x)$ 是个体 x 规范化后的适应值。可以看出, x 的扰动幅度与 $f^*(x)$ 成反比。

选择策略采用确定性的选择方案,与 $(\mu+\lambda)$ -ES选择算子不同的是,每个个体在它的邻域内随机搜索产生 λ 个个体,然后从此 $\lambda+1$ (包括此父个体本身)个个体中挑出最好个体进入下一代群体,之所以没有让群体的所有子个体进行比较,是因为我们要定位所有的峰值点,而不是只定位全局峰值点,所以选择压力不能过大。

2.4步的聚集过程基本思想仍然是基于距离的。其目的是让个体位置不要太拥挤,在每个山峰上尽量只有一个个体。过程如下:

第一步,对当前群体中的个体按从好到坏次序排序并记为集合 C_s ;

第二步,按 C_s 中的顺序依次对 C_s 中每一个个体做在当前个体的邻域内看是否有 C_s 中的个体落在其中,若有,则从 C_s 中去掉这些个体。

第三步,结束

通过上述聚集过程后的 C_s 集合中的个体就是去掉了多余的个体后剩下的个体。该思想与机器学习中的聚类分析一

个明显不同的是,这里每一个类的“核”可以动态确定的(因为个体有顺序),每个类中的最好个体(核)被保留,而聚类分析是一种无监督学习过程,没有顺序的概念。个体的表示为向量表示,不需要编码,个体之间的距离定义为欧式距离,邻域大小为 σ 。

停机条件是连续3代平均群体数目不变化,表明算法已找不到更多的山峰,停机。

4 实验结果

我们选择两个 Benchmark^[5]作为算法性能的测试。

(1) $g(z) = \frac{1}{1+|z^6-1|}$, 其中 $z \in \mathbb{C}, z=x+iy, x, y \in [-2, 2]$ 。该函数在复平面上有六个峰值点,高度都为1.0,即这六个点都是全局极值点,算法所用相关参数如下:

InitPop=100; Nc=10; $\epsilon=10\%$;

$\sigma_s=0.2$; $D_s=0.015$; $\rho=100$

图3是本算法对该测试函数的结果,从图中可以看出,尽管初始群体规模=100,但在算法结束时,由于DPEA通过聚集和新个体引入两种机制动态调整群体大小,最后只剩下6个个体,分别定位在6个山峰上,且每个山峰上只有一个个体。图4给出了算法运行10次的平均群体规模变化过程。平均在170代左右,算法收敛停机。

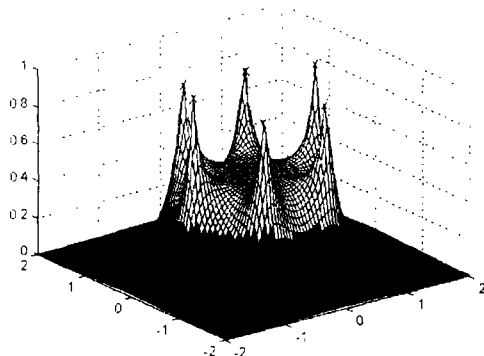


图3 利用DPEA找到了 $g(z)$ 的所有山峰

(2) $g(x, y) = x \sin(4\pi x) - y \sin(4\pi y) + 1, x, y \in [-2, 2]$ 。该函数有多个分布不均匀的局部峰值点,其中只有一个全局峰值点。

实验使用的参数如下: InitPop=100; Nc=10; $\epsilon=10\%$; $\sigma_s=0.2$; $D_s=0.015$; $\rho=100$

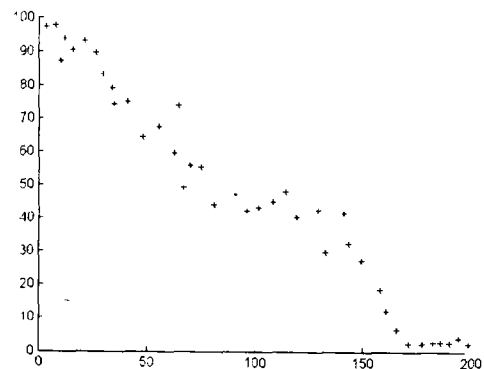


图4 求解 $g(z)$ 的DPEA运行过程中群体规模的演变

我们所找到的山峰情况如图5所示,很明显,除了个别很低的山峰没有定位到外,其他山峰都已找到,并且每个山峰上的个体数目基本上都是一个,避免了多个点定位在同一个山峰的资源浪费情况。同时,为了说明群体规模的动态调整过

程,我们分别以初始群体 InitPop=20和初始群体 InitPop=200出发分别演化,图5是运行5次的平均结果。从图6可看出两种不同趋势的群体规模演变过程。这说明最后演变定位到的山峰数目与初始群体规模无关,且最终趋向于收敛到峰值点规模大小。

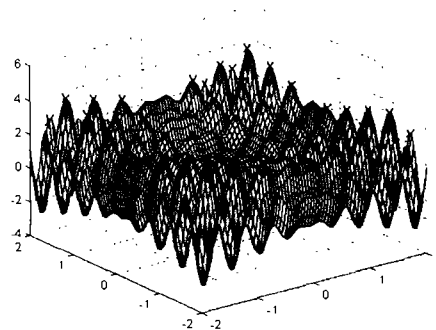


图5 求解 $g(x, y)$ 的DPEA运行结果

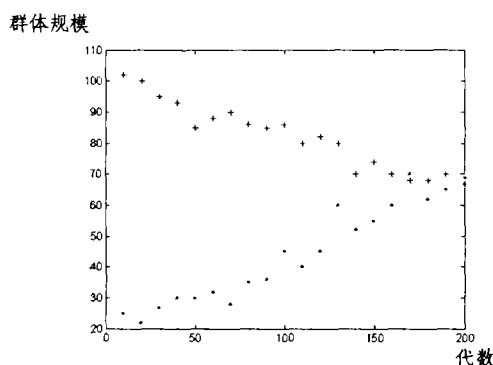


图6 两种不同的演化趋势(初始群体分别为20和200)

结论 本文首先指出了当前对多峰函数优化研究领域里的一个被忽略的内容,强调对搜索全局峰值点和局部峰值点的函数优化问题是有意义的,并指出传统的固定群体规模的演化算法框架不适合这类问题,同时给出了群体动态可调的试图寻找所有峰值点的演化算法,并给出了初步的实验结果。进一步的研究集中在:

(1)改进局部搜索机制。我们在实验中观察到,如果最好和最差个体的扰动幅度相差悬殊时,那些较差的(较低的)山峰就可能被丢弃。我们为简单起见,式(2)定义的幅度为1,因为该指数函数 $\in [0, 1]$ 。所以那些低的山峰由于经历了大的扰动而被丢弃,算法搜寻不到那些低的山峰。怎样修正式(2)及衰减参数 ρ 使得在收敛速度和定位尽可能多的山峰之间保持平衡是我们进一步实验的目的。

(2)算法的并行化和复杂性分析及改进。

参考文献

- 1 de Castro L N, von Zuben F J. Learning and Optimization Using the Clonal Selection Principle. Evolutionary Computation. IEEE Tran, 2002, 6(3)
- 2 Balazs M E, LI Jian-ping. The Effect of Distance Measure in A GA with Species Conservation [C]. In: Proc. of 2001 IEEE Congress on Evolutionary Computation, 2001. 67~74
- 3 Deb K. Multi-objective Optimization Using Evolutionary Algorithms [M]. John Wiley Sons. Ltd, 2001
- 4 Wu Zhijian, Kang Lishan, Zou Xiufen. A parallel global-local mixed evolutionary algorithm for multimodal function optimization [C]. In: Proc. of fifth Intl. conf. on algorithms and architectures for parallel processing, Beijing, 2002. 247~250
- 5 周明,孙树栋. 遗传算法原理及应用. 北京:国防工业出版社, 1999
- 6 Back T, Fogel D, Michalewicz Z. Handbook of Evolutionary Computation [M]. Bristol: Institute of Physics Publishing and New York: Oxford University Press, 1997