

基于 $(\mu + 1)$ 演化策略的多目标优化算法

周育人^{1,2}, 李元香², 王 勇², 周继香¹

(1.华南理工大学计算机科学与工程学院, 广州 510640; 武汉大学计算机学院, 武汉 430072)

摘 要: 使用 $(\mu + 1)$ 演化策略求解多目标优化问题, 利用群体中个体间的距离定义拥挤密度函数以衡量群体中个体的密集程度, 个体适应值定义为个体的Pareto强度值和拥挤密度值之和。通过对测试函数的实验, 验证了算法的可行性和有效性, 该算法具有简单、稳健等特点。

关键词: 演化算法; 多目标优化; Pareto最优解; 演化策略

Multiobjective Optimization Algorithm Based on $(\mu + 1)$ Evolutionary Strategy

ZHOU Yuren^{1,2}, LI Yuanxiang², WANG Yong², ZHOU Jixiang¹

(1.College of Computer Science and Engineering, South China University of Science and Technology, Guangzhou 510640;

2.School of Computer, Wuhan University, Wuhan 430072)

【Abstract】 This paper proposes $(\mu + 1)$ evolutionary strategy for multiobjective optimization problems. It uses crowding density to maintain a good spread of solution in the population. Sorting the distances between one point and other points in the population, the individuals crowding density is defined as the sum of the nearest and the second distances. Then it defines the fitness of the individual by combining Pareto strength and crowding density. Test results on several benchmark functions show that the approach is a simple, robust and effective method.

【Key words】 Evolutionary algorithm; multiobjective optimization; Pareto-optimal solution; Evolutionary strategy

现实世界经常提出大量的多目标优化问题(向量优化问题), 需要同时对几个相互冲突的目标进行优化。与单目标优化问题的最优解为一个点不同, 多目标优化问题的最优解一般为一个解集, 称为Pareto最优解集(Pareto-optimal Solution Set)。如果没有额外附加的信息, Pareto最优解之间相互不能比较优劣, 经典优化方法处理多目标优化问题是通过权组合将多目标优化问题转换成单目标优化问题求解, 此时找到的解实际上是一个单目标优化问题的解。演化算法是近年发展起来的一种新的全局优化算法, 最近出现了许多基于演化算法的多目标优化算法^[1-6], 即多目标演化算法(Multi-objective Evolutionary Algorithms, MOEAs), 越来越多的证据显示演化算法非常适合求解多目标优化问题, 其机理在于基于群体搜索的演化算法可并行搜索多个目标, 在搜索过程中可以将最优解保存到下一代, 这些正是求解多目标优化问题所希望的。

多目标演化算法的任务也是多目标的: (1)所求得解要尽量接近或者就是Pareto最优解; (2)所求得解集要尽量分布均匀; (3)求解过程中要防止获得的Pareto最优解丢失。本文提出使用 $(\mu + 1)$ 演化策略和拥挤密度法来求解多目标优化问题的Pareto最优解集。由于使用 $(\mu + 1)$ 演化策略, 每代只淘汰一个最差解, 淘汰压力小, Pareto最优解自动得到保存; 为了使群体中的个体分布均匀, 用拥挤密度来量化个体的拥挤程度, 以便去掉最密集点。与现有的一些较好算法, 如强度Pareto演化算法(Strength Pareto Evolutionary Algorithm, SPEA)^[1]以及Pareto档案演化策略(Pareto Archived Evolution Strategy, PAES)^[2]等算法相比较, 我们的方法具有简洁、稳健、高效等特点。

1 Pareto最优解

不失一般性, 考虑下列 m 个自变量和 n 个目标函数的多

目标函数优化问题:

$$\text{Maximize } y=f(x)=(f_1(x_1, \dots, x_m), \dots, f_n(x_1, \dots, x_m))$$

其中 $x=(x_1, \dots, x_m) \in X \subset R^m$, $y=(y_1, \dots, y_n) \in Y \subset R^n$, x 为决策(参数)向量, X 为决策空间, y 为目标向量, Y 为目标空间。

设 $a \in X$, $b \in X$, 称 a Pareto 优于 b (Pareto dominate记 $a \phi b$) 或 b 劣于 a 当且仅当 $\forall i \in \{1, \dots, n\}: f_i(a) \geq f_i(b)$ 且 $\exists j \in \{1, \dots, n\}: f_j(a) > f_j(b)$, 称 a 覆盖 b (cover记 $a \phi = b$) 当且仅当 $a \phi b$ 或 $f(a) = f(b)$ 。

显然Pareto优于关系是偏序。下面给出非劣解或Pareto最优解的定义:

定义1 设 $a \in X$, X' 为 X 的子集, 若 X' 中不存在优于 a 的向量, 则称 a 为关于子集 X' 的非劣解或关于子集 X' 的Pareto最优解, 若 a 为关于 X 的非劣解, 则 a 简称为非劣解或Pareto最优解。所有的Pareto最优解对应的目标函数值形成的区域称为均衡面(Pareto Optimal Front or Trade-off Surface)。

2 多目标进化算法

一般而言, MOEAs首先考虑的是利用群体中个体的Pareto优于关系进行演化, 以便群体向均衡面逼近。这种方法的优点在于演化结果与均衡面的凹凸无关且无需任何事先偏好信息, 但搜索空间的维数制约了这种方式的性能, 单纯的Pareto演化算法不能解决目标数目很多的优化问题, 会因

基金项目: 国家自然科学基金资助项目(69703011)

作者简介: 周育人(1965—), 男, 博士生、副教授, 主研演化计算、信号处理; 李元香, 教授、博导; 王 勇, 硕士生; 周继香, 助教

收稿日期: 2002-10-07

维数过高和均衡面过长而得不到满意解。造成这种结果的原因在于如果处理的问题空间维数过高,则有限规模的进化群体中各个体之间很难进行Pareto排序比较,有时甚至会出现所有个体皆Pareto最优解,从而无法实施正常的进化选优。所以,MOEAs的第2个必须考虑的问题是,当群体中的个体不能利用Pareto优于关系进行比较时,怎样打破个体之间的Pareto平衡关系,使比较能进行下去。文献[3]提出将决策者对多目标的偏好信息融入到搜索过程,这种方法利用了目标函数以外的信息。更自然的想法是,只利用多目标优化问题本身的信息,考虑到多目标演化算法的另一个任务是尽量使群体中的个体分布均匀,在个体的Pareto强度相同的情况下,比较个体的密集程度,这样就可以打破Pareto优于关系的平衡状态,演化搜索的选优能继续下去。下面通过对个体适应值的设置,实现上述两个步骤。

2.1 个体适应值的计算

分别定义个体的适应值的整数部分和小数部分,整数部分反映群体中个体的Pareto优于关系的强弱程度,小数部分反映群体中个体的密集程度。整数部分采用文献[1]中的强度值,小数部分采用本文提出的拥挤密度值。

设 x_i 为群体 P 中的一个个体,用 $S(x_i)$ 表示群体中Pareto劣于 x_i 的个体个数,称为 x_i 的强度值,即

$$S(x_i) = \#\{x_j | x_j \in P_i \text{ 且 } x_i \not\phi x_j\}, \text{ 其中 } \# \text{ 表示集合的基数。}$$

以强度值为基础,定义个体 x_i 的原始适应值 $R(x_i)$ (适应值的整数部分)为群体中优于 x_i 的个体的强度值和,即

$$R(x_i) = \sum_{x_j \in P_i, x_j \phi x_i} S(x_j)$$

原始适应值 $R(x_i)$ 反映了个体在群体中的Pareto优于关系的强弱程度, $R(x_i)=0$ 时,群体 P 中没有优于 x_i 的个体, x_i 为关于群体 P 的Pareto最优解; $R(x_i)$ 越大表示群体中优于 x_i 的个体越多。

虽然原始适应值 $R(x_i)$ 反映了Pareto偏序的强弱程度,但当群体中的两个个体有相同的原始适应值时,原始适应值 $R(x_i)$ 不能区分这两个个体的优劣,这种情况经常会发生,极端的情况甚至为所有的个体都具有相同的原始适应值。为了个体之间的比较能进行下去,有些演化算法转而比较群体中个体的密集程度,这时有多种不同的方法。文献[2]中使用适应性格子法(Adaptive Grid Algorithm),将群体所在的区域自适应地等分成均匀的格子,计算在每个格子中个体的数目,从而比较个体的密集程度;文献[1]中使用聚类分析法(Cluster Analysis)等等。这些方法的缺点是,算法设计比较复杂,运算量大,以下使用拥挤密度这个概念来量化群体中个体的密集程度。

设群体 P 的规模为 N , $x_i \in P$, 计算 x_i 与群体 P 中其他 $N-1$ 个个体的欧几里德距离,并从小到大排序,记为 d_i^k , $k=1,2,\dots,N-1$ 。用 $d(x_i)$ 表示群体中与 x_i 最近的两个个体与 x_i 的距离之和,即 $d(x_i) = d_i^1 + d_i^2$, $d(x_i)$ 直观上反映了个体 x_i 在群体 P 中的密集程度, $d(x_i)$ 越小, x_i 所在的位置个体越密集。

定义 $D(x_i) = \frac{1}{d(x_i) + 1}$ 称为个体 x_i 的拥挤密度,显然有 $0 < D(x_i) < 1$, 且 $D(x_i)$ 越大, x_i 所在的位置个体越密集。

最后,我们定义个体 x_i 的适应值(适应值小的个体为优): $F(x_i) = R(x_i) + D(x_i)$ (1)

显然,由式(1)所定义的个体适应值体现了以下原则:

—2—

- (1)首先比较群体中个体的Pareto优于关系强弱(整数部分);
- (2)在个体Pareto优于关系相同的情况下,比较个体的密集程度(小数部分)。

2.2 演化策略

我们采用 $(\mu+1)$ 演化策略(ES Evolution Strategy)来实现算法,编码采用实数向量编码,对于实函数的优化问题,实数编码的性能优于传统的二进制编码。由父体产生后代时,不使用杂交算子,只使用变异算子,因为直观地来看,两个(或几个)相距较远的Pareto最优个体杂交后可能会产生一个较差的个体,杂交不仅没有优势,甚至还会有劣势。一般认为自适应变异法要优于非自适应变异法,我们使用高斯变异的自适应变异法。

自适应 $(\mu+1)$ 演化策略算法描述如下:

(1)随机产生 μ 个个体组成初始群体,置 $k=1$,每个个体表示为 (x_i, η_i) , $i \in \{1, 2, \dots, \mu\}$, x_i 为 m 维自变量, η_i 为高斯变异的标准方差, x_i 和 η_i 均为 m 维向量;

(2)在 μ 个个体中任意取一个个体 (x_i, η_i) , $i \in \{1, 2, \dots, \mu\}$ 由下式产生 (x_i, η_i) 的后代 (x_i', η_i') , 对 $j=1, 2, \dots, m$,

$$x_i'(j) = x_i(j) + \eta_i(j) N_j(0, 1)$$

$$\eta_i'(j) = \eta_i(j) \exp(\tau N(0, 1) + \tau N_j(0, 1))$$

其中 $x_i(j)$ 、 $x_i'(j)$ 、 $\eta_i(j)$ 和 $\eta_i'(j)$ 分别是向量 x_i 、 x_i' 、 η_i 和 η_i' 的第 j 个分量, $N(0, 1)$ 表示一维的标准正态分布随机数, $N_j(0, 1)$ 表示标准正态分布随机数随着每一个 j 而更新,步长 τ 和 τ 分别取为 $(\sqrt{2} \sqrt{m})^{-1}$ 和 $(\sqrt{2m})^{-1}$;

(3)按式(1)计算 $(\mu+1)$ 个个体的适应值,去掉适应值最大的个体;

(4)如停机规则满足则停机,否则, $k=k+1$,并转(2)。

3 数值实验

3.1 测试函数

多目标优化问题的测试函数的设计比单目标优化问题的设计要复杂得多,文献[4]针对多目标演化算法的任务和困难设计了一系列测试函数。考虑到多目标演化算法的主要任务有两个:(1)使群体收敛到Pareto最优解集;(2)保持群体分布的均匀性。对于多目标演化算法的第1个任务,像单目标演化算法一样,将测试函数设计成具有多模型(multimodality)、欺骗性(deception)、孤立最优点(isolated optima)等特征的函数,对于多目标演化算法的第2个任务,将Pareto最优解集设计成凸的、非凸的、离散、非均匀的区域,这些都会增加演化算法完成任务的难度。

使用3个典型的测试函数来检验新算法的性能。实验函数A取自文献[5]。

$$A \text{ Minimize } T_1(x) = (f_1(x), f_2(x))$$

$$f_1(x) = 1 - \exp(-\sum_{i=1}^3 (x_i - 1/\sqrt{3})^2)$$

$$f_2(x) = 1 - \exp(-\sum_{i=1}^3 (x_i + 1/\sqrt{3})^2)$$

$$x = (x_1, x_2, x_3), x_i \in [-4, 4], i=1, 2, 3$$

问题A的Pareto最优解集为, $x_1 = x_2 = x_3 \in [-1/\sqrt{3}, 1/\sqrt{3}]$ 。

实验函数B取自文献[6]

$$B \text{ Minimize } T_2(x) = (f_1(x), f_2(x), f_3(x))$$

$$f_1(x) = 0.5(x_1^2 + x_2^2) + \sin(x_1^2 + x_2^2)$$

$$f_2(x) = (3x_1 - 2x_2 + 4)^2/8 + (x_1 - x_2 + 1)^2/27 + 15$$

$$f_3(x)=1/(x_1^2+x_2^2+1)-1.1*\exp(-x_1^2-x_2^2)$$

$$x=(x_1,x_2), x_1,x_2 \in [-3,3]$$

这个问题有3个Pareto均衡面，它们在目标空间呈现为一条非线性非对称的三维螺旋曲线。

文献[4]构造的一系列具有相同形式的测试函数，由3个函数 f_1 、 g 、 h 组成：

$$\text{Minimize } T(x)=(f_1(x_1), f_2(x))$$

$$\text{其中 } f_2(x)=g(x_2, \dots, x_m)h(f_1(x_1), g(x_2, \dots, x_m))$$

$$x=(x_1, \dots, x_m)$$

x 为 m 维向量， f_1 为第一个变量的函数， g 为剩余变量的函数。随着 x 、 f_1 、 g 、 h 的不同可以构造许多不同的测试函数，选其中1个作为实验函数 C 。

$$C \text{ 令 } T_3(x)=(f_1(x_1), f_2(x))$$

$$f_1(x_1)=x_1$$

$$g(x_2, \dots, x_m)=1+9*\sum_{i=2}^m x_i/(m-1)$$

$$h(f_1, g)=1-\sqrt{f_1/g}$$

$$m=10, x_i \in [0,1], i=1, \dots, m$$

这个问题的Pareto最优解集为 $g(x_2, \dots, x_m)=1$ ，其均衡面为凸集。

3.2 算法性能的评估

不同的多目标演化算法进行适当的比较又是一件复杂的事情，因为多目标演化算法的设计比较复杂，不同的方法难以在完全相同的条件下进行比较，对于不同算法的运算结果也没有一个统一的指标来衡量算法的好坏。尽管如此，研究者仍然设计出一些方法如直观图示和性能指标^[1,7]等等来对不同的算法比较。文献[7]中定义了刻画算法收敛性指标 γ ，文献[1]中定义了比较两个解集 X_1 、 X_2 的Pareto覆盖关系的指标 $C(X_1, X_2)$ 。

定义2 设 X_1 、 X_2 为决策空间 X 的两个子集(解集)，函数 C 将子集对 (X_1, X_2) 映射到 $[0, 1]$ 区间：

$$C(X_1, X_2)=\#\{b \in X_2; \exists a \in X_1; a \phi =b\} \#X_2$$

若 $C(X_1, X_2)=1$ ，则 X_2 中的所有解劣于或等于 X_1 中的解，即 X_2 中的任意解都被 X_1 中的解覆盖；若 $C(X_1, X_2)=0$ ，则 X_2 中的任意解都不被 X_1 中的解覆盖。

定义3 设多目标优化问题的Pareto最优解集为已知集，在Pareto最优解集对应的目标函数值形成的均衡面上均匀地取一些点(比如取500个点)，计算由算法获得的解与这些点的距离的最小值最后求所有最小值的平均值得到 γ 。

γ 刻画了解集收敛到Pareto最优集的程度，同时也包含了解集分布均匀性的信息，但是它的计算要求Pareto最优集为已知。 γ 值越小则算法的性能越好。

3.3 实验结果

新方法的实验在MATLAB中完成，参数 μ 取70，运行代数数为10000代。图1为实验函数A的目标函数的均衡面的演化结果，图2和图3分别为实验函数B的Pareto最优解集和目标空间的Pareto曲线，从图中可以看出，对于二维和三维的目标函数，新方法都较好地演化出其Pareto曲线。

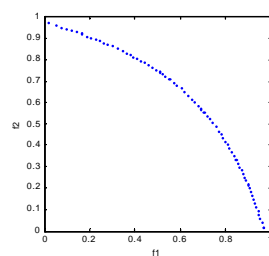


图1 函数A的Pareto曲线

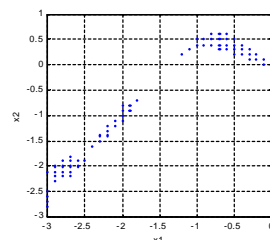


图2 函数B的Pareto最优集

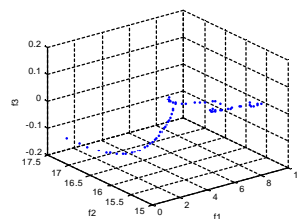


图3 函数B的Pareto曲线

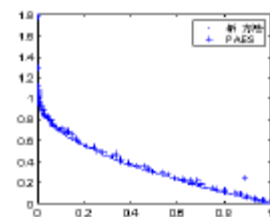


图4 函数C的Pareto曲线比较图

对测试函数 C ，将新算法的运算结果和文献[2]中PAES算法进行比较，PAES采用(1+1)-ES，档案(archive)保留精英个体(取其规模为70)，适应性格子法去掉密集的个体。将新方法和PAES各运行20次，每次10000代，计算收敛性度量 γ (见定义3)的值，比较结果如表1，图4为一次典型的运算结果比较。

表1 新方法和PAES计算收敛性度量值 γ 的比较

| | 平均值 | 标准差 |
|--------|----------|----------|
| 新方法 | 0.012635 | 0.000003 |
| PAES算法 | 0.082084 | 0.000012 |

4 小结

演化算法特别适合于求解多目标优化问题，同时，演化算法求解多目标优化问题的设计比单目标问题的设计要困难得多。现有的算法大都采用两个群体、精英保留策略、聚类分析等方法，算法的整体设计显得较复杂。本文使用 $(\mu+1)$ 演化策略求解多目标优化问题，提出了使用拥挤密度来保持群体中个体的均匀分布，将个体的Pareto强度值和拥挤密度合并到个体的适应值定义中。通过对测试函数的实验，验证了算法的可行性和有效性。

参考文献

- 1 Zitzler E, Thiele L. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Transaction on Evolutionary Computation, 1999, 3(4): 257-271
- 2 Knowles J, Corne D. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. Evolutionary Computation, 2000, 8(2): 149-172
- 3 崔逊学. 多目标协调进化算法研究. 计算机学报, 2001, (9): 979-984
- 4 Deb K. Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Function. Evolutionary Computation, 1999, (7): 205-230
- 5 Fonseca C, Fleming P. Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms (Part II): Application Example. IEEE Transactions on Systems, Man, and Cybernetics: Part A, 1998: 38-47
- 6 Veldhuizen D, Lamont G. Multiobjective Evolutionary Algorithm Test Suites. Proc. Symp. Applied Computing, San Antonio, TX, 1999: 351