第 31 卷第 4 期
2014 年 4 月

控 制 理 论 与 应 用
Control Theory & Applications

Vol. 31 No. 4
Apr. 2014

DOI: 10.7641/CTA.2014.30596

基于局部抽象凸支撑面的多模态优化算法

邓勇跃, 张贵军†
(浙江工业大学 信息工程学院, 浙江 杭州 310023)

摘要: 在基本进化算法框架下, 结合抽象凸理论, 提出一种基于局部抽象凸支撑面的多模态优化算法. 首先, 采用模型变换方法将原优化问题转变为单位单纯形约束条件下的严格递增射线凸松弛问题; 其次, 针对新生成个体的邻域信息构建局部抽象凸支撑面, 并利用局部下界知识动态识别种群模态, 从而减少替换误差, 避免出现早熟现象; 最后, 借助支撑面下降方向进一步实现模态内部的局部增强过程. 数值研究表明, 针对给定的绝大部分测试问题, 提出的算法在精度和可靠性指标方面均优于文中给出的其他算法.

关键词: 进化算法; 抽象凸; 支撑向量; 多模态优化; 下界估计

中图分类号: TP391　　**文献标识码**: A

Multimodal optimization based on local abstract convexity support hyperplanes

DENG Yong-yue, ZHANG Gui-jun†
(College of Information Engineering, Zhejiang University of Technology, Hangzhou Zhejiang 310023, China)

Abstract: In the framework of basic evolutionary algorithms, a new multimodal optimization algorithm based on local abstract convexity support hyperplanes is proposed by using the abstract convexity theory. Firstly, the original bound constrained optimization problem is converted to an increasing convex along rays (ICAR) relaxed problem over unit simplex by using the projection transformation method. Secondly, we construct the underestimate support hypeplanes with the information of trial individual neighborhood and make use of the local lower bound to identify the potential niches dynamically, thus reducing the replacement error and avoiding the premature. Finally, with the aid of descendent direction of support hyperplanes, the detected niches will be enhanced at the same time. Experiments had been performed on several benchmark functions. For most of the benchmark functions, the numerical results show the proposed algorithm is capable to provide better and more consistent performance over the existing multimodal algorithms both in accuracy and reliability.

Key words: evolutionary algorithms; abstract convexity; support vector; multimodal optimization; underestimation

1 引言(Introduction)

多模态优化是全局优化领域的一个热点研究方向. 近10年来, 在国内外引起广泛关注, 并提出了一系列有效的算法[1–3]. 多模态优化方法在工程应用领域有着广泛的应用. 如在机械设计领域, 设计者不但要找出产生最大共振的频率, 还要找出振幅在一定阈值以上的所有其他频率, 以消除共振在工程上产生的负面影响[4]. 在蛋白质结构预测领域, 由于力场模型的复杂性和不精确性, 计算获得的全局最优解不一定是最稳定的物理结构, 而某些局部最优解才是实际需要的稳态结构[5], 因此要求尽可能确定问题的所有全局最优解和多个质量较高的局部极值解.

一般来讲, 多模态优化算法需要重点解决两个问题: 1) 生境判断问题, 即如何判别新发现的极值点是否与已求出的极值点属于同一生境; 2) 不同生境保持问题, 即如何将不同较优生境在后续的进化过程中稳定的保存下来. 传统的方法是在遗传算法(GA)[6–7]、差分进化(DE)[8–9]、粒子群优化(PSO)[10]等群体进化算法的基础上融入排挤策略[11]、物种保存策略[12]、清洗策略[13]和适应度分享策略[14], 以更好的保存多个极值解. 然而小生境半径的确定问题限制了这些方法的有效应用, 如物种保存策略中的物种半径、清洗策略中的清洗半径以及适应度分享中共享半径等等. 截至目前, 还没有一个有效的方法可以确定合适的小生境半径. 相比于随机优化方法, 确定性全局优化以丰富的数学理论基础, 在给定的误差范围内确保得到

收稿日期: 2013−06−14; 录用日期: 2013−12−31.
†通信作者. E-mail: zgj@zjut.edu.cn; Tel.: +86 13958125042.
基金项目: 国家自然科学基金资助项目(61075062, 61379020); 浙江省自然科学基金资助项目(LY13F030008); 浙江省重中之重学科开放基金资助项目(20120811); 杭州市产学研合作资助项目(20131631E31).

问题的全局最优解, 同时可以获取相应的上下界信息. 目前, 比较典型的方法有凸分析[15]、双层规划[16]、分支定界法[17]、抽象凸分析[18]等. 然而, 确定性优化方法极高的时间复杂度和空间复杂度制约了其在大规模问题中的应用.

为了解决多模态优化中的生境半径确定难问题以及确定性优化方法中代价极高的计算复杂度, 本文权衡两种算法缺陷, 并综合利用两者优势, 提出了一种基于局部抽象凸支撑面的多模态优化算法(multi-modal optimization based on local abstract convexity support hyperplanes, LAC). 在传统进化算法的框架下, 算法将目标优化问题转换为单位单纯形约束条件下的严格递增射线凸松弛问题. 在更新环节利用新产生个体邻域信息构建下界低估支撑面, 通过引入主导个体思想, 寻找主导个体所属的支撑向量, 并利用该支撑向量的下降方向指导新个体更加高效的替换现有种群个体, 并减少替换误差, 避免出现早熟现象, 从而产生质量更高的新个体. 在进化算法中融入局部抽象凸下界估计思想不仅提高了进化算法的可靠性, 而且有效降低了确定性全局优化算法的计算复杂度.

2 研究动机(Research motivation)

传统的进化算法在更新种群的时候具有盲目性. 如在排挤策略中, 新产生的个体只替换离其距离最近的个体, 其一个潜在的问题是trial个体与被替换的个体可能不在同一个模态里面, 导致替换错误[2]. 在物种保存策略中, 通常会指定一个物种半径, 如果新产生的个体落在某个半径之内, 则替换该物种里面最差的个体, 否则替换整个种群中最差的个体. 替换的精确度往往取决于半径选择是否合适, 如果选取了一个不合适的半径, 则有可能导致算法在运行过程中某些有效模态的丢失, 致使算法收敛于问题的局部极值解[12]. 适应度分享策略同样存在适应度共享半径的合理选取问题[14].

鉴于上述的这些问题, Floudas等学者考虑将随机优化算法与确定性算法αBB[19–20]结合起来用于求解多模态优化问题[26], 其中αBB通过构建下界凸包络, 并以不断收紧的下界逼近目标函数, 从而求得问题的全局最优解. 然而, α的求解是一个及其富有挑战性的工作, 且没有一个有效的方法可以廉价的确定α值. 相对而言, Rubinov A等提出的抽象凸理论[18,21–23]将目标优化问题转换为组合优化问题, 通过枚举方法可以快速得到下界估计值, 从而高效的构建目标函数下界支撑面.

为降低计算复杂度, 本文并没有采用在整个可行域空间构建下界低估面来逼近目标函数的传统思路, 而是在传统进化算法的框架下, 运用抽象凸理论局部构建下界支撑面来引导新个体的替换过程, 使算法的进化以一个更加合理的方向进行. 一方面进化算法的整体框架可以提升算法的执行效率, 另一方面, 由于仅在局部构建下界支撑面, 算法并没有产生很高的计算复杂度.

3 理论基础(Theoretical basis)

抽象凸理论为确定性全局优化提供了有力的理论支撑. 凸分析的一个重要结论是任一个凸函数是它一系列仿射弱函数的上包络. 抽象凸则泛化了仿射弱函数的概念, 引入次梯度这一有力的分析工具, 通过不断构建支撑向量来逼近目标函数, 并求得全局最优解.

本文中, 构建一个有效的下界支撑面, 并快速枚举相应的支撑函数极值解是实现LAC算法的关键. 构建合理的支撑向量有助于降低新个体的替换误差, 而高效的枚举下界支撑点有利于实现算法有效的更新过程, 提高算法的收敛速度.

定义 1 设$f_{\rm x}(\lambda)\equiv\{f(\lambda x)|x\in\mathbb{R}_+^{N+1},\lambda\in(0,+\infty)\}$, 如果$f:\mathbb{R}_+^{N+1}\to\mathbb{R}$在$\lambda$射线方向为凸函数, 则称$f:\mathbb{R}_+^{N+1}\to\mathbb{R}$为射线凸函数(convex along rays, CAR); 进一步的, $\forall x,y\in\mathbb{R}_+^{N+1}$, 当$x\geqslant y$时满足$f(x)\geqslant f(y)$, 则称$f:\mathbb{R}_+^{N+1}\to\mathbb{R}$为递增射线凸函数(increasing convex along rays, ICAR); 特别地, 当$x>y$时满足$f(x)>f(y)$, 那么称$f:\mathbb{R}_+^{N+1}\to\mathbb{R}$为严格递增射线凸函数(strictly increasing convex along rays, SICAR)[21].

定义 2 若函数f满足如下两个条件:

$$\forall x,y\in\mathbb{R}_+^n,\ x\geqslant y\Rightarrow f(x)\geqslant f(y), \tag{1}$$

$$\forall x\in\mathbb{R}_+^n,\ \forall\lambda\in\mathbb{R}_{++}: f(\lambda x)=\lambda f(x), \tag{2}$$

那么称函数$f:\mathbb{R}_+^n\to\mathbb{R}$为正齐次递增函数(increasing positively Homogeneous functions of degree one, IPH)[22].

引理 1 如果$f:S\to\mathbb{R}$为Lipschitz正函数, $y\in S=\{y\in\mathbb{R}_+^n:\sum\limits_{i=1}^n y_i=1\}$且满足

$$L=\inf\{k:k\geqslant\frac{|f(x)-f(y)|}{\|x-y\|_1},\ x\neq y,\ \forall x,y\in S\}.$$

取$p\geqslant\max\{1,2L/C\}$, 其中:

$$\|\cdot\|_1=\sum_{i=1}^n|x_i|,\ C=\min_{x\in S}f(x),$$

则

$$g(x)=\begin{cases}f(\dfrac{x}{\sum\limits_{i=1}^n x_i})(\sum\limits_{i=1}^n x_i)^p, & x\neq 0,\ x\in\mathbb{R}_+^n,\\ 0, & x=0,\ x\in\mathbb{R}_+^n\end{cases}$$

为射线凸递增函数.

证 详细证明过程见参考文献[21].

定理 1 令f是一个ICAR函数, $y\in\mathbb{R}_+^{N+1}\backslash\{0\}$,

则存在$l = u/y,\ u \in \partial f_y(1)$, 使得

$$\min_{i\in I} l_i x_i - \min_{i\in I} l_i y_i \leqslant f(x) - f(y) \tag{3}$$

对于任意$x \in \mathbb{R}_+^{N+1}$成立.

证 基于凸分析理论可知$\partial f_y(\lambda), \forall\lambda \in (0,\infty)$非空, 且$\partial f_y(\lambda) \in [f_y^-(\lambda), f_y^+(\lambda)]$, 其中:

$$f_y^-(\lambda) = \lim_{\beta\to 0^-}\frac{f_y(\lambda) - f_y(\lambda-\beta)}{\beta} = \lim_{\beta\to 0^-}\frac{\lambda f(y) - (\lambda-\beta)f(y)}{\beta} = f(y), \tag{4}$$

$$f_y^+(\lambda) = \lim_{\beta\to 0^+}\frac{f_y(\lambda+\beta) - f_y(\lambda)}{\beta} = \lim_{\beta\to 0^+}\frac{(\lambda+\beta)f(y) - \lambda f(y)}{\beta} = f(y). \tag{5}$$

故$\partial f_y(\lambda)|_{\lambda=1} = \partial f_y(1) = f(y)$. 因此

$$\min_{i\in I} l_i x_i - \min_{i\in I} l_i y_i = \min_{i\in I}\frac{f(y)}{y_i}x_i - \min_{i\in I}\frac{f(y)}{y_i}y_i = f(y)\min_{i\in I}\frac{x_i}{y_i} - f(y). \tag{6}$$

记

$$\lambda = \min\{\frac{x_1}{y_1}, \frac{x_2}{y_2}, \cdots, \frac{x_{N+1}}{y_{N+1}}\}, \tag{7}$$

则对于$\forall t \in \{1, 2, \cdots, N+1\}$, $\frac{x_t}{y_t} \geqslant \lambda$成立.

根据ICAR的定义,

$$x_t \geqslant \lambda y_t \Leftrightarrow f(x_t) \geqslant f(\lambda y_t) \Leftrightarrow f(x) \geqslant f(\lambda y). \tag{8}$$

因此, $\min\limits_{i\in I} l_i x_i - \min\limits_{i\in I} l_i y_i = \lambda f(y) - f(y) \leqslant f(x) - f(y)$. 故存在$l \in \mathbb{R}_+^{N+1}$, 使得

$$\min_{i\in I} l_i x_i - \min_{i\in I} l_i y_i \leqslant f(x) - f(y) \tag{9}$$

对于任意$x \in \mathbb{R}_+^{N+1}$成立. 命题得证.

基于定理1, 本文又有如下推论:

推论 1 设$y^1, y^2, \cdots, y^K \in S$, 则

$$H^K(x) = \max_{k=1,\cdots,K} h^k(x) = \max_{k=1,\cdots,K}\min_{i\in I} l_i^k x_i$$

为$f: S \to \mathbb{R}_+$的支撑函数族, 则

$$H^K(x) \leqslant f(x),\ \forall x \in S, \tag{10}$$

$$H^K(x) = f(x),\ \forall x \in \{y^1, y^2, \cdots, y^K\}. \tag{11}$$

证 定理1中, $\min\limits_{i\in I} l_i y_i = f(y)$, 故不等式(9)进一步简略为$\min\limits_{i\in I} l_i x_i \leqslant f(x)$, 记$h(x) = \min\limits_{i\in I} l_i x_i$, 则$h(x) \leqslant f(x)$, 即$\forall x^k, k \in \{1, 2, \cdots, K\}$, 有

$$h(x^k) = h^k(x) \leqslant f(x^k),$$

故不难得到

$$H^K(x) = \max_{k=1,\cdots,K} h^k(x) \leqslant f(x),\ \forall x \in S.$$

又因为

$$h^k(x) = \min_{i\in I} l_i^k x_i = f(x^k)\min_{i\in I}\frac{x_i^k}{y_i}, \tag{12}$$

故当$x^k = y$时, $\lambda = \min\limits_{i\in I} x_i^k/y_i = 1$, 故

$$h^k(y) = f(y),\ \forall x \in \{y^1, y^2, \cdots, y^K\},$$
$$H^K(x) = \max_{k=1,\cdots,K} h^k(x) = f(x),$$

即$H^K(x) = f(x)$. 证毕.

4 局部抽象凸支撑面优化算法(LAC algorithm)

4.1 基本思想(Basic idea)

整个算法是在进化算法的框架下进行的. 本文提出的算法只应用于进化算法的更新环节, 其他环节与标准的进化算法一致. 为了使trial个体的进化更有方向性, 提高种群个体替换的准确率, 本文在抽象凸理论的基础上, 通过构建trail个体附近种群个体的下界支撑面, 并利用下界信息判断trial个体落在哪个支撑面上, 进而引导trial个体的更新.

以图1的二维问题为例, 假设C为trial个体, 寻找离其最近的两个种群个体A和B, 构建下界支撑面, 求得下界支撑函数极值点$D(x_\mathrm{u}, d(x_\mathrm{u}))$. 判断C在B主导的支撑面上, 故B为主导个体, 且C的适应度优于B的适应度, 则B被C个体取代. 为了提升算法收敛速度, 本文继续判断支撑函数极值点D在目标函数曲面上对应的点$D'(x_\mathrm{u}, f(x_\mathrm{u}))$, 因为$D'$的适应度值优于C，故C被$D'$个体取代, 这样完成一次种群的更新过程.

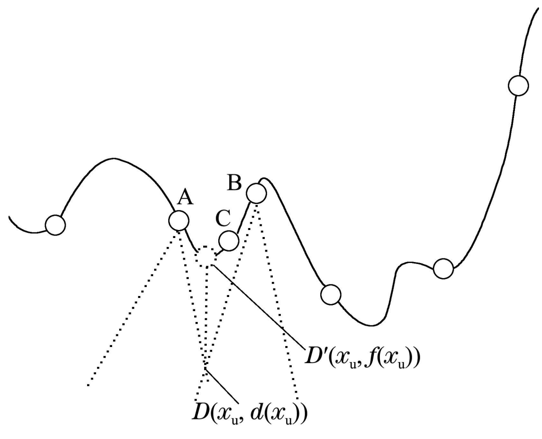

图 1 LAC中trial个体更新过程示意图

Fig. 1 Demo of trial individual updating procedure in LAC

4.2 模型转换(Model transformation)

考虑一个N维的极小值优化问题:

$$\min f(x), x \in [a, b] \subset \mathbb{R}^N, \tag{13}$$

其中: $x = (x_1, x_2, \cdots, x_N)^\mathrm{T}$为边界约束可行域空间$[a, b]$上的$N$维连续优化变量, 而常向量$a = (a_1, a_2, \cdots, a_N)^\mathrm{T}$和$b = (b_1, b_2, \cdots, b_N)^\mathrm{T}$分别表示可行域空间的上界和下界, $f(\cdot)$为定义在边界约束可行域

$[a,b]$上的目标优化函数, 在给出的可行域空间内可能包含多个全局最优解和一系列局部极小值解.

为适用LAC算法, 需将矩形可行域映射为单位单纯形空间. 故将原优化变量$x=(x_1,x_2,\cdots,x_N)^{\mathrm{T}}$作如下线性变换:

$$\begin{cases} {x'}_i \equiv (x_i-a_i)/\sum\limits_{i=1}^{N}(b_i-a_i), \\ {x'}_{N+1} \equiv 1-\sum\limits_{i=1}^{N}{x'}_i, \end{cases} \tag{14}$$

其中: $i=1,2,\cdots,N$, $x'_1,x'_2,\cdots,x'_{N+1}$为线性转换变量. 显然

$$\sum_{i=1}^{N}(b_i-a_i)>0,\ x_i-a_i\geqslant 0,\ i=1,2,\cdots,N+1,$$

易知$x'_i\geqslant 0$, $i=1,2,\cdots,N+1$, $x'_{N+1}\geqslant 0$, 即

$$x'\equiv(x'_1,x'_2,\cdots,x'_{N+1})^{\mathrm{T}}\in S\equiv \{x'\in\mathbb{R}_+^{N+1}:\sum_{i=1}^{N+1}{x'}_i=1\},$$

其中: $\mathbb{R}_+^{N+1}\equiv\{x'=(x'_1,x'_2,\cdots,x'_{N+1})|x'_i\geqslant 0,\ i=1,2,\cdots,N+1\}$, S表示单位单纯形空间, 图2给出了一个将二维定义域转换为单位单纯形空间的示意图.

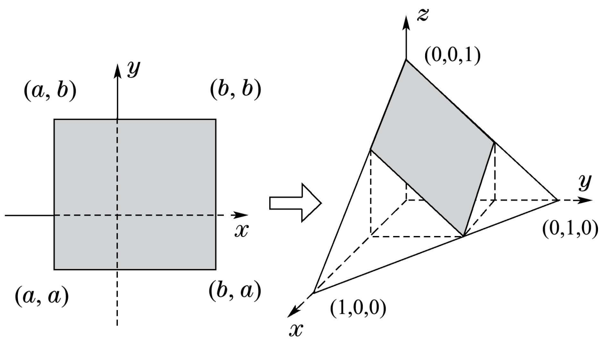

图 2　二维定义域映射为单位单纯形

Fig. 2　The 2-dimension domain projecting to unit simplex

将式(14)反变换, 可得到

$$x_i=x'_i\sum_{i=1}^{N}(b_i-a_i)+a_i,\ i=1,2,\cdots,N.$$

将上式代入式(13), 可得$\min g(x')$, $x'\in S\subset\mathbb{R}_+^{N+1}$, 其中$g(\cdot)$为定义在$N+1$维单位单纯形空间$S$上的目标函数.

4.3　快速枚举下界值(Enumerating underestimate values quickly)

所有ICAR函数均可以转换为IPH函数, 实际上IPH函数是ICAR函数的特殊情形. 引理1中, 当$C\geqslant 2L$时, 有$p\geqslant 1$, 取$p=1$, 则ICAR函数转换为自由度为1的IPH函数. 本文中, 所有测试函数的定义域已利用模型转换公式映射到单位单纯形空间

$$S_1=\{x\in\mathbb{R}^{N+1}:x_i\geqslant 0,\ \sum_{i=1}^{N+1}x_i=1\}.$$

由引理1可知, 对于一给定的Lipschitz常数L, 可以通过加上一个足够大的常数C, 将测试函数转换为IPH函数. 假定有K个支撑函数, 基于给定的点$(x^k,f(x^k))$, $k=1,\cdots,K$, 则目标函数f的下界估计可由式(15)给出[22], 由定理1以及推论1可知, 式(12)可以构建一个有效的下界低估函数:

$$H^K(x)=\max_{k\leqslant K}\min_{i=1,\cdots,N+1}l_i^k x_i, \tag{15}$$

第k个支撑函数定义如下:

$$h^k(x)=\min_{i=1,\cdots,N+1}l_i^k x_i= f(x^k)\min\{\frac{x_1}{x_1^k},\cdots,\frac{x_{N+1}}{x_{N+1}^k}\},$$

其中

$$l^k=(\frac{f(x^k)}{x_1^k},\frac{f(x^k)}{x_2^k},\cdots,\frac{f(x^k)}{x_{N+1}^k}) \tag{16}$$

称为支撑向量集. 考虑一个含有K个支撑向量的集合, $S=\{l^k\}_{k=1}^K$, $l^k\in\mathbb{R}_+^{N+1}$, 令I表示$\{1,2,\cdots,N+1\}$, 则满足条件关系式(17)–(18)的$N+1$个支撑向量集$L=\{l^{k_1},l^{k_2},\cdots,l^{k_{N+1}}\}$, 称为一个有效的支撑矩阵[22], 其中$\varLambda^K$表示所有支撑向量的集合.

$$\forall i,j\in I,\ i\neq j: l_i^{k_i}<l_i^{k_j}, \tag{17}$$

$$\forall v\in\varLambda^K\backslash L,\ \exists i\in I: l_i^{k_i}\geqslant v_i. \tag{18}$$

假设一个有效的支撑矩阵如下:

$$L=\begin{pmatrix} l_1^{k_1} & l_2^{k_1} & \cdots & l_{N+1}^{k_1} \\ l_1^{k_2} & l_2^{k_2} & \cdots & l_{N+1}^{k_2} \\ \vdots & \vdots & \ddots & \vdots \\ l_1^{k_{N+1}} & l_2^{k_{N+1}} & \cdots & l_{N+1}^{k_{N+1}} \end{pmatrix}, \tag{19}$$

则根据该支撑矩阵可以计算得到相应的下界支撑函数极值解[22]:

$$x_{\min}(L)=\mathrm{diag}(L)/\mathrm{trace}(L), \tag{20}$$

$$d(L)=H^K(x_{\min})=\mathrm{trace}(L)^{-1}. \tag{21}$$

4.4　算法设计(Algorithm design)

4.4.1　构建虚拟支撑面(Virtual support hypeplanes construction)

假设问题维数为N, 本文需要构建$N+1$个单位单纯形的虚拟边界支撑面, 确保给出问题的最小有效下界估计. 同时, 根据进化算法产生的新个体, 对种群个体按欧几里得距离升序排列, 获取$N+1$个离新个体最近的种群个体, 由式(16)构建新个体的支撑面.

4.4.2　判断主导个体(Dominator judgement)

当虚拟支撑面构建好之后, 需要判断trial个体落在哪个支撑面上, 相应的支撑面所属的种群个体称为主导个体(dominator).

在传统进化算法中, 通过选择、交叉、变异之后产

生一个新个体trial, 然后利用主导个体判断算法寻找当前的主导个体. 判断步骤如下: 首先寻找与trial最近的$N+1$个种群个体, 通过式(16)分别计算各个维度下的支撑向量, 构建一个下界支撑面. 然后, 算法求解trial个体在各支撑向量上的值value, 并记录下界最大估计值所对应的支撑向量位置location, 最后找到该支撑向量对应的种群个体即为主导个体.

具体的判断算法如下:

主导个体判断算法:

输入: 新个体trial, 支撑向量sv;

输出: 主导个体dominator;

初始化estimateValue为0; $x^k[i]$保存$sv[k]$对应位置上的元素. location保存trial个体所属的支撑向量, value保存相应的低估值;

```
for k = 1 to N + 1
   for i = 1 to N + 1
      计算f(sv[k]),
      计算value = f(sv[k]) * trial[i]/x^k[i];
      if estimateValue < value
         estimateValue = value;
         location = k;
      end
   end
end
由sv[location]得到主导个体;
返回主导个体dominator;
结束.
```

4.4.3 算法描述(Algorithm description)

1) 参数初始化: 设置参数C, 计算可行域S顶点处$x^1, x^2, \cdots, x^{N+1}$目标函数值, $f(x^1), f(x^2), \cdots, f(x^{N+1})$, 计算初始支撑向量集合

$$\Lambda^{N+1} = \{l^1, l^2, \cdots, l^{N+1}\},$$
$$L = \{l^1, l^2, \cdots, l^{N+1}\},$$

根据式(19)构建虚拟下界支撑面;

2) 新产生的个体trail, 寻找离其最近的$N+1$个个体, 生成支撑向量sv, 构建下界支撑面;

3) 利用主导个体判断算法判断trial落在哪个支撑面上, 支撑向量所属的种群个体称之为dominator个体;

4) 如果dominator个体存在,

a) 如果$f(\text{trial}) < f(\text{dominator})$, 则以trail个体替换dominator;

b) 计算下界支撑函数极值解x_{u}, 如果

$$f(x_{\text{u}}) < f(\text{trial}),$$

则用x_{u}替换trial;

5) 否则, 替换离trial个体最近的种群个体.

注 1 算法只需构建trial个体对应的下界支撑面, 在算法每一轮结束后, 新构建的下界支撑面将删除, 只保留虚拟支撑面, 降低了空间复杂度;

注 2 C值应当设置的足够大, 以确保目标函数为IPH函数, 本文中设置C为100;

注 3 根据式(20)计算下界支撑函数极值解x_{u}.

4.5 复杂度分析(Complexity analysis)

本文算法排序平均时间复杂度为$\text{O}(\log(N+1))$, 构建下界支撑面时间复杂度为$\text{O}(N+1)$, 主导个体判断时间复杂度为$\text{O}(N+1)$, 故整体算法时间复杂度为$\text{O}((N+1)\log(N+1))$. 相比于传统的算法, 清洗策略中, 算法的平均复杂度为$\text{O}(C\cdot N^2)$, 其中C为清洗半径内的种群容量; 排挤策略的平均复杂度为$\text{O}(N)$; 适应度分享策略的平均复杂度为$\text{O}(N^2)$; 物种保存策略的平均复杂度介于$\text{O}(N)$和$\text{O}(N^2)$之间. 因此本文算法在引入局部抽象凸的基础上并未显著增加算法的复杂性.

5 数值仿真(Numerical experiments)

5.1 测试问题(Benchmark functions)

由于算法是在进化算法框架下进行, 本文在数值仿真部分选用差分进化算法. 限于篇幅, 差分进化算法的整体算法流程可以参考文献[8–9].

为综合比较算法性能, 本文选取了近年来提出且比较有竞争力的多模态优化算法TSC2[24], 以及4种粒子群优化算法(PSO)的变种[25]r2pso, r3pso, r2pso-lhc, r3pso-lhc, 选取8个典型的测试函数(见表1)问题作为对比. 各算法独立运行30次, 种群容量为100, 差分进化算法中的参数设置为$CR=0.1$, $F=0.5$. 为保证算法性能比较的客观性, 实验以30,000次评价函数为算法终止条件.

表 1 8个多模态优化问题

Table 1 8 multimodal benchmark functions

函数	尺寸	优化数
Shubert	2	18
Rastrigin(2D)	2	25
Rastrigin(10D)	10	1
Griewank	5	1
PP5	10	4
Ursem F1	2	2
Ursem F3	2	5
Ursem F4	2	5

5.2 评价指标(Evaluation indicators)

实验采用如下评价指标:

1) 成功率. 探测到的峰值点与峰值点总数的比值, 其中, 如果探测到的峰值点与真实峰值点的欧几里得

距离误差在0.1之内, 可以认为已经找到目标函数的峰值点.

2) 峰值精度. 所有真实峰值点与已探测到离其最近点适应度值的绝对值之和, 具体定义见式(22).

$$\text{peak acc.} = \sum_{i=1}^{\#\text{peak}\,s} |f(\text{peak}_i) - f(x)|. \tag{22}$$

3) 距离精度. 如果某些峰值点有相同的值, 且各峰值点相距很近, 则即使只探测到其中的一个或几个峰值, 峰值精度也会很高. 为了预防潜在的误差, 论文引进了距离精度. 其计算方式与峰值精度类似, 只需要把适应度函数替换为欧几里得距离即可.

实验选取3个高维问题Griewank, Rastrigin(10D), PP5, 4个多模态优化问题, 其中Shubert函数包含最优值为-186.731的18个全局最优解, 2维的Rastrigin包含一个全局最优解和24个局部极值解, 10维的Rastrigin具有一个全局最优解, 10维的PP5包含4个全局最优解, Ursem F1包含一个全局最优解和一个次最优解, Ursem F3包含一个全局最优解和4个次最优解, Ursem F4包含一个全局最优解和4个位于可行域边界的4个次最优解.

从表2易知, 对于Shubert函数, LAC算法在30次独立运行中均可以找到所有的全局最优解, PSO变种算法成功率降至0.85左右, 而TSC2在30次运行中平均成功率仅为0.13. Griewank函数具有较大的可行域空间, 以及较高的维度, 本文所列算法均表现出了较低的性能, TSC2及PSO的变种在30次独立运行中均未获得成功, LAC算法的成功率也降至0.27. 而针对10维的Rastrigin, LAC算法在30次独立运行中均以较高的精度探测到了全局最优解. 对于Ursem F1, Ursem F3, LAC算法依然维持在100%的成功率, 其中TSC2在Ursem F3中也表现出了良好的性能. 而r2pso则分别将至0.6, 0.53. 至于Ursem F4, LAC虽然运行效果不如r2pso-lhc, r3pso-lhc, 然而只表现出微弱的弱势.

在峰值精度方面, 表3给出了实验运行的结果.

表 2 30次独立运行各测试函数成功率, 平均(最优)

Table 2 The success rate of benchmark functions in 30 independent runs, average (best)

Benchmark	LAC	TSC2	r2pso	r3pso	r2pso-lhc	r3pso-lhc
Shubert	1(1)	0.13(0.61)	0.85(0.94)	0.84(1)	0.85(1)	0.84(1)
Rastrigin(2D)	0.92(1)	0.88(0.96)	0.65(0.76)	0.52(0.68)	0.67(0.84)	0.56(0.76)
Rastrigin(10D)	1(1)	0(0)	0(0)	0(0)	0(0)	0(0)
Griewank	0.27(1)	0(0)	0(0)	0(0)	0(0)	0(0)
PP5	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)
Ursem F1	1(1)	0.76(1)	0.96(1)	0.60(1)	1(1)	1(1)
Ursem F3	1(1)	1(1)	0.88(1)	0.53(0.8)	0.92(1)	0.75(1)
Ursem F4	0.31(0.6)	0.44(1)	0.36(0.4)	0.30(0.4)	0.4(0.6)	0.4(0.6)

表 3 30次独立运行各测试函数的峰值精度, 平均(最优)

Table 3 The peak accuracy of benchmark functions in 30 independent runs, average (best)

Benchmark	LAC	TSC2	r2pso	r3pso	r2pso-lhc	r3pso-lhc
Shubert	2.46E − 04 (8.32E − 05)	12.08 (4.56)	1.11 (3.85E − 02)	1.75 (1.53E − 03)	9.07E − 01 (2.65E − 03)	5.98E − 01 (5.24E − 03)
Rastrigin(2D)	6.41E − 01 (2.44E − 03)	6.56E − 01 (4.52E − 02)	7.13 (4.24)	14.7 (8.92)	7.44 (3.94)	11.8 (5.18)
Rastrigin(10D)	1.19E − 05 (3.57E − 06)	9.14 (3.58)	16.6 (5.79)	17.1 (9.05)	17.7 (11.8)	16.6 (7.81)
Griewank	8.46 (6.94E − 04)	82.70 (23.92)	4477.60 (262.21)	3390.45 (177.36)	4262.25 (539.18)	3657.85 (268.04)
PP5	80.75 (56.45)	245.12 (180.65)	4800.68 (625.55)	4780.50 (597.75)	4528.65 (585.24)	3947.83 (425.65)
Ursem F1	1.32E − 06 (4.75E − 08)	1.33E − 02 (1.53E − 03)	5.43E − 03 (2.19E − 07)	1.81E − 02 (2.01E − 08)	2.90E − 07 (9.59E − 09)	4.53E − 07 (6.82E − 09)
Ursem F3	2.51E − 05 (3.20E − 06)	8.04E − 05 (5.80E − 06)	6.85E − 02 (2.23E − 03)	4.41E − 01 (8.72E − 03)	1.17E − 01 (1.80E − 03)	6.46E − 01 (2.77E − 03)
Ursem F4	2.84 (5.37E − 02)	2.36E − 01 (1.26E − 02)	1.75 (4.32E − 01)	2.84 (5.37E − 02)	1.48E − 01 (1.25E − 02)	1.65 (2.87E − 02)

从表3中不难发现, 在30次独立运行中, LAC在Shubert, Rastrigin, Griewank, 以及Ursem F3中表现出了明显的优势. 在其余测试函数中, LAC也表现出了平稳的性能, 如PP5, 虽然由于函数较为复杂, 给定的算法未能在给定的精度内求得问题最优解, 但是在峰值精度方面, 与其他算法相比, 仍表现出了较好的优势. 至于Ursem F4, LAC略逊于TSC2以及PSO变种算法, LAC在处理峰值点位于边界的情形有一定的缺陷. 在距离精度性能指标中(见表4), 除Ursem F1和Ursem F4, 在其余测试函数中, LAC算法均表现出了较高的性能.

另外, 为了验证提出的算法与现有算法相比是否有显著优势, 本文引入了两种非参数假设检验t–test和Wilcoxon test. 这里, 两种假设检验的显著性水平均取为0.05. 从表5和表6可以看出, 除了Ursem F4, LAC表现的不尽如人意之外, 相比与其他算法, 优势均十分明显, 表明LAC算法是一种有效的多模态算法.

表 4 30次独立运行各测试函数的距离精度, 平均(最优)

Table 4 The distance accuracy of benchmark functions in 30 independent runs, average (best)

Benchmark	LAC	TSC2	r2pso	r3pso	r2pso-lhc	r3pso-lhc
Shubert	5.02E − 02 (1.56E − 02)	11.68 (6.57)	1.69 (3.72E − 01)	3.59 (2.84E − 01)	2.48 (1.87E − 01)	3.24 (2.59E − 01)
Rastrigin(2D)	8.60E − 01 (2.29E − 01)	1.66 (0.85)	7.77 (5.06)	12.37 (5.66)	7.74 (4.25)	11.4 (5.55)
Rastrigin(10D)	3.36E − 03 (1.89E − 03)	2.99 (1.89)	4.02 (2.41)	4.17 (3.01)	4.24 (3.43)	4.08 (2.79)
Griewank	1.49 (2.63E − 02)	8.83 (4.89)	62.76 (16.19)	54.42 (13.31)	60.20 (23.25)	56.84 (16.34)
PP5	32.56 (26.45)	45.23 (32.88)	100.35 (52.32)	85.45 (49.55)	72.35 (45.67)	68.76 (35.12)
Ursem F1	1.19E − 03 (2.88E − 04)	1.06E − 01 (3.92E − 02)	5.38E − 02 (5.47E − 04)	3.75E − 01 (1.88E − 04)	2.67E − 03 (1.29E − 04)	1.79E − 03 (1.16E − 04)
Ursem F3	7.92E − 03 (2.13E − 03)	1.48E − 02 (4.35E − 03)	2.95E − 01 (9.60E − 02)	1.66 (5.91E − 01)	3.18E − 01 (8.62E − 02)	1.14 (1.01E − 01)
Ursem F4	4.22 (1.70)	7.64E − 01 (1.85E − 01)	2.07 (1.01)	4.22 (1.70)	2.52 (9.05E − 01)	3.49 (1.20)

表 5 各测试函数的峰值精度假设检验, t-test (Wilcoxon test)

Table 5 t-test and Wilcoxon test on peak accuracy presented as "t-test (Wilcoxon test)"

Benchmark	TSC2	r2pso	r3pso	r2pso-lhc	r3pso-lhc
Shubert	9.48E − 14 (3.01E − 11)	1.28E − 09 (3.02E − 11)	0.18 (3.02E − 11)	6.28E − 08 (3.02E − 11)	9.80E − 06 (3.02E − 11)
Rastrigin(2D)	9.27E − 01 (2.98E − 01)	3.45E − 17 (2.09E − 11)	6.90E − 19 (2.09E − 11)	4.97E − 19 (2.09E − 11)	2.95E − 18 (2.09E − 11)
Rastrigin(10D)	1.48E − 18 (3.02E − 11)	3.79E − 17 (3.02E − 11)	2.22E − 21 (3.02E − 11)	1.55E − 20 (3.02E − 12)	1.07E − 18 (3.01E − 11)
Griewank	2.76E − 10 (2.61E − 10)	2.89E − 08 (3.02E − 11)	6.55E − 08 (3.02 − 11)	2.62E − 07 (3.02E − 11)	1.76E − 08 (3.02E − 11)
PP5	3.45E − 05 (3.54E − 07)	2.43E − 03 (1.55E − 05)	3.25E − 08 (7.45E − 10)	5.46E − 08 (5.47E − 10)	8.28E − 07 (3.75E − 08)
Ursem F1	4.85E − 07 (3.02E − 11)	2.50E − 02 (1.61E − 06)	3.35E − 04 (6.52E − 09)	1.26E − 02 (4.91E − 06)	5.26E − 02 (1.16E − 05)
Ursem F3	5.43E − 06 (6.04E − 07)	1.90E − 02 (3.02E − 11)	1.65E − 06 (3.02E − 11)	2.46E − 02 (3.02E − 11)	1.49E − 06 (3.02E − 11)
Ursem F4	3.59E − 06 (4.12E − 06)	0.41 (0.76)	0.10 (0.54)	1.08E − 06 (6.52E − 08)	0.71 (7.24E − 02)

表 6 各测试函数的距离精度假设检验, t-test (Wilcoxon test)

Table 6 t-test and Wilcoxon test on distance accuracy presented as “t-test (Wilcoxon test)”

Benchmark	TSC2	r2pso	r3pso	r2pso-lhc	r3pso-lhc
Shubert	$7.84\mathrm{E}-19$ $(3.02\mathrm{E}-11)$	$8.63\mathrm{E}-12$ $(3.02\mathrm{E}-11)$	$7.18\mathrm{E}-06$ $(3.02\mathrm{E}-11)$	$4.78\mathrm{E}-07$ $(3.02\mathrm{E}-11)$	$3.49\mathrm{E}-08$ $(3.02\mathrm{E}-11)$
Rastrigin(2D)	$3.90\mathrm{E}-05$ $(2.59\mathrm{E}-05)$	$4.17\mathrm{E}-20$ $(2.59\mathrm{E}-11)$	$1.23\mathrm{E}-20$ $(2.59\mathrm{E}-11)$	$7.08\mathrm{E}-21$ $(2.59\mathrm{E}-11)$	$3.88\mathrm{E}-20$ $(2.59\mathrm{E}-11)$
Rastrigin(10D)	$4.50\mathrm{E}-26$ $(3.02\mathrm{E}-11)$	$1.38\mathrm{E}-24$ $(3.02\mathrm{E}-11)$	$4.62\mathrm{E}-29$ $(3.02\mathrm{E}-11)$	$2.43\mathrm{E}-27$ $(3.02\mathrm{E}-11)$	$4.75\mathrm{E}-26$ $(3.02\mathrm{E}-11)$
Griewank	$7.28\mathrm{E}-13$ $(2.61\mathrm{E}-10)$	$1.24\mathrm{E}-14$ $(3.02\mathrm{E}-11)$	$4.29\mathrm{E}-14$ $(3.02\mathrm{E}-11)$	$2.54\mathrm{E}-13$ $(3.02\mathrm{E}-11)$	$9.01\mathrm{E}-15$ $3.02\mathrm{E}-11$
PP5	$5.35\mathrm{E}-08$ $(2.49\mathrm{E}-09)$	$6.26\mathrm{E}-10$ $(5.25\mathrm{E}-10)$	$6.34\mathrm{E}-12$ $(3.02\mathrm{E}-11)$	$8.78\mathrm{E}-09$ $(2.84\mathrm{E}-10)$	$9.25\mathrm{E}-09$ $(1.07\mathrm{E}-11)$
Ursem F1	$6.27\mathrm{E}-13$ $(3.02\mathrm{E}-11)$	$5.38\mathrm{E}-04$ $(6.28\mathrm{E}-06)$	$3.64\mathrm{E}-06$ $(5.57\mathrm{E}-10)$	0.19 $(8.0\mathrm{E}-02)$	0.36 $(3.64\mathrm{E}-02)$
Ursem F3	$2.48\mathrm{E}-06$ $(8.83\mathrm{E}-07)$	$3.37\mathrm{E}-09$ $(3.02\mathrm{E}-11)$	$2.27\mathrm{E}-12$ $(3.02\mathrm{E}-11)$	$2.51\mathrm{E}-05$ $(3.02\mathrm{E}-11)$	$4.37\mathrm{E}-10$ $(3.02\mathrm{E}-11)$
Ursem F4	$6.93\mathrm{E}-05$ $(6.20\mathrm{E}-04)$	$4.0\mathrm{E}-03$ $(1.44\mathrm{E}-02)$	$3.38\mathrm{E}-09$ $(1.41\mathrm{E}-09)$	$1.91\mathrm{E}-05$ $(1.52\mathrm{E}-05)$	$7.99\mathrm{E}-07$ $(2.60\mathrm{E}-08)$

6 结论(Conclusions)

论文在传统进化算法的基础上, 在进化算法的更新环节引入抽象凸理论, 构建新产生个体的有效下界估计, 通过判断trial个体落在支撑面的位置, 来确定其替换策略. 相比于传统的抽象凸分析方法, 本文算法避开了N–叉树的构建和复杂的节点查询操作. 事实上, 下界支撑面的引入是为了更好的引导trial个体的替换操作. 因此, 本文只需在trial个体处构建下界支撑面, 一旦更新操作完成, 新构建的支撑面随即被删除, 保存下界估计点的内存空间也随之释放. 实验数据表明, LAC是一种有效的多模态优化算法, 该算法不仅使得已探测到的模态可以在后期运行过程中平稳保存下来, 而且利用支撑向量下降方向信息可以起到模态内部的局部增强作用, 有效提升了算法的可靠性和收敛速度. 而且, 更新过程是进化算法的通用操作, 所以本文提出的策略也可以广泛的应用到遗传算法、蚁群系统、粒子群等群体优化算法中.

参考文献(References):

[1] DAS S, SUGANTHAN P N. Differential evolution-a survey of the state-of-theart [J]. *IEEE Transactions on Evolutionary Computation*, 2011, 15(1): 4 – 31.

[2] DAS S, MAITY S, QU B Y, et al. Real-parameter evolutionary multimodal optimization – a survey of the state-of-the-art [J]. *Swarm and Evolutionary Computation*, 2011, 1(2): 71 – 88.

[3] FLOUDAS C A, GOUNARIS C E. A review of recent advances in global optimization [J]. *Journal of Global Optimization*, 2009, 45(1): 3 – 38.

[4] BEASLEY D, BULL D R, MARTIN R D. A sequential niche technique for multimodal function optimization [J]. *Evolutionary Computation*, 1993, 1(2): 101 – 125.

[5] BRADLEY P, MISURA K M S, BAKER D. Toward high-resolution de novo structure prediction for small proteins [J]. *Science*, 2005, 309(4742): 1868 – 1871.

[6] GOLDBERG D E. *Genetic Algorithms in Search, Optimization, and Machine Learning* [M]. New York: Addison-Wesley, 1989.

[7] 巩敦卫, 陈健, 孙晓燕. 新的基于相似度估计个体适应值的交互式遗传算法 [J]. 控制理论与应用, 2013, 30(5): 558 – 566.
(GONG Dunwei, CHEN Jian, SUN Xiaoyan. Novel interactive genetic algorithm for estimating individual fitness based on similarity [J]. *Control Theory & Applications*, 2013, 30(5): 558 – 566.)

[8] STORN R, PRICE K V. Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces [J]. *Journal of Global Optimization*, 1995, 11(4): 341 – 359.

[9] 张贵军, 王信波, 俞立, 等. 求解高维多模优化问题的自适应差分进化算法 [J]. 控制理论与应用, 2008, 25(5): 862 – 866.
(ZHANG Guijun, WANG Xinbo, YU Li, et al. Adaptive differential evolution for high-dimension multimodal optimization problems [J]. *Control Theory & Applications*, 2008, 25(5): 862 – 866.)

[10] MATHEW M N. A new gradient based particle swarm optimization algorithm for accurate computation of global minimum [J]. *Applied Soft Computing*, 2012, 12(1): 353 – 359.

[11] THOMSEN R. Multimodal optimization using crowding-based differential evolution [C] //*Proceedings of IEEE International Conference on Evolutionary Computation*. Oregon: Portland, 2004: 1382 – 1389.

[12] LI J P, BALAZS M E, PARKS G T, et al. A species conserving genetic algorithm for multimodal function optimization [J]. *Evolutionary Computation*, 2002, 10(3): 207 – 234.

[13] PETROWSKI A. A clearing procedure as a niching method for genetic algorithms [C] //*Proceedings of IEEE International Conference on Evolutionary Computation*. Nagoya: Japan, 1996: 798 – 803.

[14] CIOPPA A D, STEFANO C D, MARCELLI A. Where are the niches? Dynamic fitness sharing [J]. *IEEE Transactions on Evolutionary Computation*, 2007, 11(4): 453 – 464.

[15] BERTSEKAS D, NEDIC A. *Convex Analysis and Optimization* [M]. [s.l.]: Athena Scientific, 2003.

[16] DEMPE S. *Foundations of Bilevel of Programming, Volume 61 of Nonconvex Optimization and Its Application* [M]. Boston: Kluwer Academic Publisher, 2002.

[17] CHEN D S, BATSON R G, DANG Y. *Applied Integer Programming: Modeling and Solution* [M]. New York: John Wiley & Sons, 2010.

[18] RUBINOV A M. *Abstract Convexity and Global Optimization, Nonconvex Optimization and Its Applications* [M]. Dordrecht, Netherlands: Kluwer, 2000.

[19] ASJIMAN C S, DALLWIG S, FLOUDAS C A, et al. A global optimization method, αBB, for general twice-differentiable constrained NLPs-I Theoretical advances [J]. *Computers Chemistry Engineering*, 1998, 22(9): 1137 – 1158.

[20] ASJIMAN C S, ANDROULAKIS I P, FLOUDAS C A. A global optimization method, αBB, for general twice-differentiable constrained NLPs-II: Implementation and computational results [J]. *Computers Chemistry Engineering*, 1998, 22(9): 1159 – 1179.

[21] RUBINOV A, ANDRAMONOV M. Lipschitz programming via increasing convex-along-rays functions [J]. *Optimization Methods and Software*, 1999, 10(6): 763 – 781.

[22] BELIAKOV G. Geometry and combinatorics of the cutting angle method, optimization [J]. *Optimization*, 2003, 52(4): 379 – 394.

[23] BELIAKOV G, LIM K F. Challenges of continuous global optimization in molecular structure prediction [J]. *European Journal of Operational Research*, 2007, 181(3): 1198 – 1213.

[24] STOEAN C, PREUSS M, STOEAN R, et al. Multimodal optimization by means of a topological species conservation algorithm [J]. *IEEE Transactions on Evolutionary Computation*, 2010, 14(6): 842 – 864.

[25] LI X D. Niching without niching parameters: particle swarm optimization using a ring topology [J]. *IEEE Transactions on Evolutionary Computation*, 2010, 14(1): 150 – 169.

[26] KLEPEIS J L, PIEJA M J, FLOUDAS C A. Hybrid global optimization algorithms for protein structure prediction: alternating hybrids [J]. *Biophysical Journal*, 2003, 84(2): 869 – 882.

作者简介:

邓勇跃 (1987–), 男, 硕士研究生, 主要研究领域为智能优化, E-mail: zjykdyy@gmail.com;

张贵军 (1974–), 男, 博士, 教授, 主要研究领域为智能信息处理、全局优化理论及算法设计、生物信息学, E-mail: zgj@zjut.edu.cn.