

# 一种基于跳出局部最优机制的自适应响应面方法\*

申富伟<sup>1</sup>, 白富生<sup>2</sup>

(1. 重庆师范大学数学科学学院, 重庆 401221; 2. 重庆国家应用数学中心, 重庆 401331)

**摘要:**【目的】针对昂贵黑箱函数优化问题提出一种能够更好平衡局部搜索和全局搜索的响应面方法。【方法】对径向基函数响应面约束优化算法进行了改进,首先增加了判断陷入局部最优的机制,然后引入填充函数方法辅助选择新采样点,进而利用新采样点跳出局部最优。【结果】在37个公开的全局优化算法测试问题上的数值实验表明,本文提出的基于填充函数跳出局部最优机制的自适应响应面算法能够收敛到更好的近似全局最优解。【结论】基于填充函数方法的局部最优解跳出机制能够提升算法的全局搜索能力。

**关键词:**填充函数;昂贵黑箱函数;响应面方法;径向基函数

**中图分类号:** O224

**文献标志码:** A

**文章编号:** 1672-6693(2023)02-0018-10

黑箱函数指没有显式数学表达式的函数。黑箱优化问题指带有黑箱函数的优化问题。求解黑箱优化问题时,无法直接使用黑箱函数的内在性质,如梯度信息、凸性等。此外,黑箱函数的估值可能会涉及大规模仿真程序的执行或者理化实验的实施,因此会产生大量的时间或经济成本。一般地,称目标函数估值代价高昂的黑箱函数为昂贵黑箱优化问题。具有盒子约束的昂贵黑箱优化问题表示如下:

$$\begin{aligned} \min f(x), \\ \text{s. t. } x \in \Omega. \end{aligned}$$

其中:目标函数  $f(x)$  是黑箱函数,箱子集  $\Omega = [a, b] \in \mathbf{R}^n$  是函数定义域。黑箱函数广泛地存在于工程实践当中,如部分理化实验的数值结果可以视为实验参数的黑箱函数,在给定数据集的情况下,深度神经网络代价函数的最小损失值可以看成超参数的黑箱函数等<sup>[1]</sup>。特别地,昂贵黑箱优化是指针对估值代价比较高的黑箱函数优化问题。这类目标函数的函数值计算往往会涉及复杂物理化学实验的实施或者大规模仿真程序的执行,因此产生的高昂的时间成本、经济成本等。

昂贵黑箱优化问题来源广泛,在工程技术等诸多领域都有涉及。文献[2]解决的地下水修复问题涉及一个偏微分方程组求解,该方程组描述了地下水的污染和净化之间的动态关系。在该问题中,地下水的净化程度可以看成是井水注入速度的昂贵黑箱函数,该黑箱函数优化需要求解一个偏微分方程组,求解代价高。文献[3]为了最大程度节省桥梁焊接费用,用黑箱函数优化方法解决了桥梁的最优几何尺寸设计问题。

昂贵黑箱优化问题的困难在于目标函数没有明确的数学表达式,无法使用函数的梯度和凸性信息。因此,传统的基于梯度的优化算法如牛顿法等不能直接用来求解黑箱优化问题。除此之外,一些仅仅需要函数值信息的无导数优化算法如差分进化算法、自适应网格搜索算法<sup>[4]</sup>等往往依赖于大量的函数估值才能获得较高精度的解。而昂贵黑箱优化问题需要在尽可能少的函数估值的情况下得到问题的较好的近似解。

目前,基于响应面的算法是处理昂贵黑箱优化问题的主流算法之一,众多专家和学者对它展开了深入而又广泛的研究。这类算法性能优劣关键在于新估值点的选取,学者们根据采样点的选取机制的不同进行了相关研究。例如,1998年, Jones 等人<sup>[5]</sup>提出了 EGO 算法,该算法使用克里金插值作为响应面,并通过最大期望改善策略来得到下一个采样点;2001年, Gutmann<sup>[6]</sup>提出了一种基于径向基函数响应面的 Gutmann-RBF 全局优化算

\* 收稿日期:2022-10-05 修回日期:2022-12-17 网络出版时间:2023-04-21T09:12

资助项目:国家自然科学基金面上项目(No. 11871128);重庆市技术创新与应用发展专项(No. cstc2021jscx-jbgsX0001);重庆市教育委员会科学技术研究计划重点项目(No. KJZD-K202114801)

第一作者简介:申富伟,男,研究方向为最优化理论与算法,E-mail:2020110510064@stu.cqnu.edu.cn;通信作者:白富生,男,教授,博士,E-mail:fsbai@cqnu.edu.cn

网络出版地址:https://kns.cnki.net/kcms/detail/50.1165.N.20230420.1518.030.html

法,该算法把响应面函数崎岖度最低的点作为下一个采样点;2005年,Regis等人<sup>[7]</sup>提出了径向基函数响应面约束优化(optimization using response surfaces)算法(CORS算法),利用距离约束来控制下一个候选点与已有采样点的距离,并且引入一个模式向量 $\theta$ 来平衡距离约束和极小化响应面准则。文献[7]的数值结果表明,CORS算法优于同时期的其他同类算法;2015年Jie等人<sup>[8]</sup>提出了AMGO算法。该算法将距离约束与响应面的值做乘积来选取新的采样点,并通过引入一个向量 $\tau$ 来平衡距离约束和极小化响应面准则;2022年,文献[9]使用线性回归作为黑箱函数在某个邻域内的代理模型来预测下降方向,进而得到新采样点的邻域回归算法(NRO算法)。该文献数值结果表明,NRO算法优于EGO算法。

黑箱优化的关键在于对定义域的探索与利用的平衡。探索指的是全局搜索,当算法陷入局部极小值时,进行全局搜索,探索那些未被采样的区域,以获得更低的盆谷。利用则是指局部搜索,当算法搜索到全局极小值附近时,在该处进行更加充分的采样,得到该点附近更好的采样点。良好的算法应当根据算法最优解下降程度,自适应的选择进行探索还是利用。

目前大多数方法还没有自适应的平衡全局搜索和局部搜索的机制。CORS算法通过循环使用一个权重向量 $\theta$ 里的值来平衡全局搜索和局部搜索,AMGO算法通过向量 $\tau$ 来平衡全局搜索和局部搜索。以上两种方法对利用和探索的平衡方式过于机械,无法根据算法运行特点自适应地选择平衡方案。

填充函数方法(filled function method,FFM)<sup>[10-12]</sup>是一种有效的全局优化方法。填充函数方法通过构造辅助函数,并通过求解一个子优化问题来获得新解。由填充函数特性,该解位于原目标函数更好盆谷里。本文通过在引入填充函数方法来自适应平衡局部搜索和全局搜索,利用填充函数辅助选点提高算法的全局搜索能力。

## 1 预备理论

响应面方法是一大类方法的统称,这类方法使用带有显式数学表达式的函数在某种最佳逼近意义下近似代替黑箱函数。在昂贵黑箱优化问题当中,基于响应面的优化方法的一般会优于非响应面方法。

选择具有良好的对黑箱函数的近似表示能力的响应面能够提升算法性能。常见的插值、拟合模型都可以作为响应面模型,比如二次插值模型<sup>[13]</sup>、径向基函数插值模型<sup>[7,14-16]</sup>、克里金插值模型<sup>[5,17]</sup>、支持向量回归<sup>[18-19]</sup>、深度神经网络<sup>[1]</sup>等。除了使用上述单一响应面之外,工程应用中还常使用多种响应面相结合的形式,如使用凸组合的形式将多种单一响应面组合成混合响应面。采用这类响应面通常可以得到更好的数值结果<sup>[20]</sup>。然而,这类混合相应面的近似表示能力与组合系数有关,如何设置组合系数是应用中的难点。本文旨在研究填充函数寻优机制对平衡局部搜索和全局搜索所起的作用。为方便起见,使用形式较为简单的径向基函数插值模型。

### 1.1 带线性尾项的径向基函数插值

形如 $\varphi(\|x-x_0\|)$ 这种形式的函数叫径向基函数。其中 $x_0$ 为径向基函数的中心,函数 $\varphi:\mathbf{R}^+\rightarrow\mathbf{R}$ 。函数 $\varphi$ 的常见形式如表1所示。

表 1 常见径向基函数类型

Tab. 1 Types of common radial basis function

形式	中文名称	英文名称
$\varphi(r)=r$	线性径向基函数	Linear
$\varphi(r)=r^3$	三次径向基函数	Cubic
$\varphi(r)=r^2\log(r)$	薄板样条径向基函数	Thin plate spline
$\varphi(r)=e^{-\gamma r^2}$	高斯径向基函数	Gaussian
$\varphi(r)=\sqrt{r^2+\gamma^2}$	多二次径向基函数	Multiquadratic

注: $\gamma$ 为参数,范围为正实数集。

径向基函数插值模型为:

$$s(\mathbf{x}) = \sum_{i=1}^K \lambda_i \varphi(\|\mathbf{x} - \mathbf{x}_i\|) + \boldsymbol{\beta}^T \mathbf{x} + \alpha \quad (1)$$

式中: $\mathbf{x}_i, i=1,2,\dots,K$ 为插值点, $\boldsymbol{\beta}^T \mathbf{x} + \alpha$ 为线性尾项。利用待定系数法可以得到模型(1)的解,即是各项系数 $\lambda_i \in \mathbf{R}, i=1,2,\dots,K, \boldsymbol{\beta} \in \mathbf{R}^n, \alpha \in \mathbf{R}$ 。

设已有插值点集为:

$$\mathcal{B} = \{(\mathbf{x}_j, f(\mathbf{x}_j)) \mid \mathbf{x}_j \in \mathbf{R}^n, f(\mathbf{x}_j) \in \mathbf{R}, j = 1, 2, \dots, K\}. \quad (2)$$

通过将插值点集  $\mathcal{B}$  带入模型(1)中,得到如下线性系统:

$$\begin{bmatrix} \Phi & P \\ P^T & \mathbf{0}_1 \end{bmatrix} \begin{bmatrix} \beta \\ \alpha \end{bmatrix} = \begin{bmatrix} F \\ \mathbf{0}_2 \end{bmatrix}. \quad (3)$$

系统(3)中,  $\mathbf{0}_1$  是  $n+1$  阶的全零方阵,  $\mathbf{0}_2$  是  $n+1$  行的全零列向量,

$$\Phi = \begin{bmatrix} \varphi(\|\mathbf{x}_1 - \mathbf{x}_1\|) & \cdots & \varphi(\|\mathbf{x}_1 - \mathbf{x}_K\|) \\ \varphi(\|\mathbf{x}_2 - \mathbf{x}_1\|) & \cdots & \varphi(\|\mathbf{x}_2 - \mathbf{x}_K\|) \\ \vdots & & \vdots \\ \varphi(\|\mathbf{x}_K - \mathbf{x}_1\|) & \cdots & \varphi(\|\mathbf{x}_K - \mathbf{x}_K\|) \end{bmatrix}, P = \begin{bmatrix} \mathbf{x}_1^T & 1 \\ \mathbf{x}_2^T & 1 \\ \vdots & \vdots \\ \mathbf{x}_K^T & 1 \end{bmatrix}, \Lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_K \end{bmatrix}, F = \begin{bmatrix} f(\mathbf{x}_1) \\ f(\mathbf{x}_2) \\ \vdots \\ f(\mathbf{x}_K) \end{bmatrix}. \quad (4)$$

线性系统(4)存在唯一解当且仅当矩阵  $P$  列满秩<sup>[21]</sup>.

## 1.2 昂贵黑箱优化算法框架

昂贵黑箱优化算法一般是一系列迭代步构成,通过不断增加新的采样点更新响应面模型,使得响应面与黑箱函数在最优解附近具有较高的近似程度。

黑箱优化算法一般包含以下几个步骤:

步骤 1,在黑箱函数定义域上采样获得初始采样点,并对其进行函数估值;

步骤 2,利用已估值点集构造响应面;

步骤 3,在一定采样机制之下,选取采样点并进行函数估值;

步骤 4,若达到终止条件,则退出程序;否则,转步骤 2。

在步骤 1 中,由于拉丁超立方采样算法具有良好的空间填充性<sup>[22]</sup>,因此常被用来作为优化算法初始化采样技术。

步骤 2 和步骤 3 是算法的关键环节,通过构造子优化问题来得到新采样点。通常构造子优化问题需要利用响应面信息和新采样点和已估值点集的距离。在响应面与黑箱函数具有一定逼近程度的意义下,算法更加倾向于选择响应面的极小值作为新采样点。然而,若只考虑响应面信息,则算法极易陷入局部最优解。因此,需要用新采样点到已估值点集的具有一定的距离,这样有利于跳出局部最优解,提升了算法的全局寻优能力。

步骤 4 是算法的终止准则,对于昂贵黑箱函数而言,工程应用上总希望通过尽可能少的黑箱函数估值得到具有较高近似程度的近似最优解。因此,大多昂贵黑箱优化算法通过定义一个最大估值次数  $M$ ,并判断迭代是否达到最大估值次数作为终止准则。

## 1.3 CORS 算法

CORS 算法是由文献[7]提出出来的一种针对昂贵黑箱优化问题的优化算法。该方法通过响应面在一定的距离约束之下最小化响应面的思想来构造子优化问题。在每一个迭代步,新点选取子优化问题构造方式如下:

$$\mathbf{x}_{\text{new}} = \arg \min \{f(\mathbf{x}) \mid d(\mathbf{x}, \varphi) \geq \theta_i \Delta, \mathbf{x} \in \Omega\}. \quad (5)$$

式(5)中  $\varphi = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K\} \subset \Omega$  是已估值点构成的集合,  $d(\mathbf{x}, \varphi)$  表示点  $\mathbf{x}$  到已估值点集的距离,  $\Delta$  为距离的阈值,定义如式(6):

$$d(\mathbf{x}, \varphi) = \min \{\|\mathbf{y} - \mathbf{x}\| \mid \mathbf{y} \in \varphi\}, \Delta = \max_{\mathbf{x} \in \Omega} d(\mathbf{x}, \varphi). \quad (6)$$

距离约束通过模式向量  $\theta$  来体现。  $\theta$  代表了新点选择子优化问题对全局搜索和局部搜索的平衡,它的取值由事先给定,文献[7]取  $\theta = [0.90; 0.75; 0.25; 0.05; 0.03; 0]$ 。  $\theta$  中的元素从 1 到 0 单调递减,  $\theta_i$  的值越大,新估值点离已采样点集越远,这意味着算法越倾向于做全局搜索;反之,  $\theta_i$  越小,则表示算法越倾向于做局部搜索。在算法迭代过程当中周期性地循环调用权重模式向量  $\theta$ ,即调用的索引  $i$  与当前迭代次数  $K$  有如下关系:

$$i = \begin{cases} K \bmod m, K \bmod l \neq 0 \\ m, K \bmod l = 0 \end{cases}.$$

式中,  $m$  表示搜索模式向量  $\theta$  中的分量个数。

CORS 算法对全局搜索和局部搜索的平衡来源于模式向量。然而对模式向量的使用仅仅是随着迭代步循环调用。这一种选择模式过于机械,不能按照算法进行的阶段自适应地判断是进行局部搜索还是全局搜索。

#### 1.4 填充函数算法

填充函数方法是一种重要的全局优化方法,它通过在目标函数局部极小值处构造辅助函数来跳出局部最优解。填充函数最早是由 Ge 等人<sup>[10]</sup>提出,此后有很多学者对此方法进行了改进,构造了数值性能更加完善的填充函数。

为了便于对填充函数的阐述,此处引入文献[10]提出的盆谷和峰的概念。

盆谷是用来描述优化问题局部极小值位置的一个重要概念,它是根据局部极小值所定义的连通域。关于目标函数  $f(x)$ ,在它的一个局部极小值  $x^*$  处的盆谷  $B(x^*)$  定义为:若  $x \in B(x^*)$ ,则以  $x$  为起点执行最速下降算法收敛到  $x^*$ ,若  $x \notin B(x^*)$ ,则以  $x$  为起点执行最速下降算法无法收敛到  $x^*$ 。

峰是与盆谷相对应的概念,它是由目标函数的局部极大值  $x^*$  所定义的一个连通域。峰  $H(x^*)$  关于函数  $f(x)$  在局部极大值  $x^*$  处的定义为: $H(x^*)$  是  $-f(x)$  在  $x^*$  处的盆谷。

填充函数方法需要构造一个辅助函数,辅助函数在  $x^*$  局部邻域是一个峰,而原目标函数在  $x^*$  局部邻域是一个盆谷。从函数空间位置上看,辅助函数就像是将原目标函数位于  $x^*$  处的盆谷填高了一样,因此辅助函数也叫填充函数。

1990 年,文献[10]提出的填充函数形式如下:

$$P(x; r, \rho) = 1/[r + f(x)] \exp(-\|x - x^*\|^2/\rho^2), \quad (7)$$

式中  $r, \rho$  是参数。由于式(7)的填充函数使用指数函数构造,当  $\rho$  太小或者  $\|x - x^*\|$  太大时,  $P(x; r, \rho)$  和  $\nabla P(x; r, \rho)$  变得很小,从而产生假的平稳点。此外,  $p(x, r, \rho)$  只在线上存在极小值点,实现算法很困难。因此,一些在数值性能上更加优良的填充函数相继被提了出来。

为了克服上述缺点,文献[11]提出了如下数学表达式的填充函数:

$$P(x, x^*, p, a) = -1/\arctan[f(x) - f(x^*)] - a\|x - x^*\|^p, \quad (8)$$

式中:  $a(a > 0)$  充分大,  $p \in \mathbf{Z}$  是一个常数。式(8)避免指数函数的使用,具有优良的数学性质,且文献[11]数值实验表明,该填充函数具有较好数值性能。

填充函数方法求解优化问题一般包含两个迭代阶段,一是极小化阶段,二是填充阶段,

在极小化阶段,利用局部优化算法求解优化问题:

$$x_{\text{new}} = \arg \min\{f(x) \mid x \in \Omega\}. \quad (9)$$

得到一个  $f(x)$  的一个局部极小值  $x^*$ 。

在填充阶段,在  $x^*$  处构建填充函数  $P(x, x^*, p, a)$ ,然后利用局部优化算法求解问题:

$$\min\{P(x, x^*, p, a) \mid x \in \Omega\} \quad (10)$$

位于  $x^*$  附近的一个局部极小值点  $x'$ 。在满足一定的条件下,  $x'$  位于  $f(x)$  的相对于  $x^*$  更好的盆谷里<sup>[11]</sup>。最后,在  $x'$  处对原函数  $f(x)$  执行局部优化算法求解问题(9),收敛到  $x^{**}$ ,按照填充函数理论,  $x^{**}$  优于  $x^*$ 。

本文利用式(8)的填充函数作为局部最优解跳出机制引入到 CORS 算法当中。在算法陷入局部最优时,使用填充函数方法辅助选择新的采样点,提升了算法的全局搜索能力。

## 2 基于填充函数跳出局部最优机制的自适应响应面算法

如何有效平衡局部搜索和全局搜索是提高全局优化算法效率的关键。CORS 算法完全依赖  $\theta$  在给定的模式向量中循环取值来平衡全局搜索和局部搜索,并没有考虑到算法迭代过程中出现的具体情况。例如,在当前采样点已经落入黑箱目标函数的某个包含全局极小点或局部极小点的盆谷里时,下个采样点应该进一步探索当前采样点的附近,这样可以更有效地刻画黑箱函数的重要特征,从而更快找到问题的近似全局最优点。但是,按照  $\theta$  的循环取值决定的下一个采样点可能是离当前采样点很远的点,因此这种机械地平衡局部搜索和全局搜索的机制不能对应采取的搜索类型做及时调整,导致算法效率受到影响。

对于没有任何假设的黑箱函数而言,响应面仅仅是在函数值上的近似,一阶信息与黑箱函数可能有较大误差。因此,通常无法使用响应面的平稳点来判断算法是否陷入到局部最优解。在数值实验上,若算法经过多次

迭代,找到的最优函数值仍未得到下降,则此时算法很有可能陷入到局部最优解。本文记录最优函数值连续未下降累计次数,若该次数达到最大次数  $T$ ,就有理由认为算法陷入到局部最优解。

本文针对 CORS 算法中机械地循环调用模式向量  $\theta$  中的分量以选择局部搜索和全局搜索的机制进行了改进,给出了自适应选择搜索类型的机制。通过记录算法迭代中最优函数值连续未下降次数,来选择是否使用填充函数寻优机制跳出局部最优,更好地平衡局部搜索和全局搜索。相比 CORS 算法平衡局部最优和全局最优的方式,本文算法在执行过程中,能够根据最优函数值下降情况判断采样点是否陷入局部最优。在陷入局部最优时,利用填充函数辅助选择采样点,以跳出响应面的当前局部最优解,进行全局搜索,从而提升搜索效率。

以下给出本文算法伪代码。

输入:黑箱函数  $f(x)$ ,初始点数量  $n_0$ ,最大估值次数  $M$ ,最优函数值下降失败最大次数  $T$ ;

输出:最优解  $x_{\text{best}}, f(x_{\text{best}})$ 。

步骤 1,(初始化采样)初始化采样得  $\varphi = \{x_i\}_{i=1}^{n_0}$ ,并对  $\varphi$  进行估值,记录当前最优值  $f_{\text{best}}, n_{\text{fail}} \leftarrow 0, n \leftarrow n_0$ ;

步骤 2,构造响应面;

步骤 3,(CORS 选点)求解子优化问题(5)得到新采样点  $x_{\text{new}}, \varphi \leftarrow \varphi \cup \{x_{\text{new}}\}, n \leftarrow n + 1$ ;

步骤 4,对  $x_{\text{new}}$  进行估值得到  $f(x_{\text{new}})$ ,若  $f(x_{\text{new}}) < f_{\text{best}}$ ,则  $f_{\text{best}} \leftarrow f(x_{\text{new}}), x_{\text{best}} \leftarrow x_{\text{new}}, n_{\text{fail}} \leftarrow 0$ ;反之,  $n_{\text{fail}} \leftarrow n_{\text{fail}} + 1$ ;

步骤 5,(填充函数选点)若  $n_{\text{fail}} = T$ ,转步骤 5.1,否则,转步骤 6。

步骤 5.1,  $n_{\text{fail}} \leftarrow 0$ ;

步骤 5.2,在  $x_{\text{new}}$  处构造填充函数  $P(x, x^*, p, a)$ ;

步骤 5.3,以  $x_{\text{new}}$  为中心,产生一个扰动点  $x_0$  为起点,启动局部优化算法求解问题(10),得到  $x'_{\text{new}}$ ;

步骤 5.4,以  $x'_{\text{new}}$  处求解优化问题(9),获得  $x^*_{\text{new}}$ ;

步骤 5.5,对  $x^*_{\text{new}}$  进行估值得到  $f(x^*_{\text{new}}), \varphi \leftarrow \varphi \cup \{x^*_{\text{new}}\}$ ,若  $f(x^*_{\text{new}}) < f_{\text{best}}$ ,则  $f_{\text{best}} \leftarrow f(x^*_{\text{new}}), x_{\text{best}} \leftarrow x^*_{\text{new}}, n \leftarrow n + 1$ ;

步骤 6,(终止判断)若  $n \geq M$ ,算法结束,否则,转步骤 2。

步骤 1 初始化采样旨在从定义域中随机选取  $n_0$  个初始解。在步骤 1 中,还没有构建响应面,无法使用响应面评估采样点的质量。在该阶段,良好的采样算法应该使得定义域中的每个解都等概率的被选取。由于采样技术拉丁超立方采样技术<sup>[22]</sup>具有良好的空间填充性,本文使用这种采样技术进行初始化采样。

步骤 2 是利用采样点集及黑箱函数值信息构造线性系统(3),进而获得径向基函数插值模型。步骤 3 是利用 CORS 机制选取采样点,并更新采样点集。步骤 4 是对新采样点进行黑箱函数估值,记录最优解未更新次数,即是最优解未下降次数。若产生了更优的采样点  $x_{\text{new}}$ ,则将  $n_{\text{fail}}$  重新赋值为 0,否则记录数加 1。

步骤 5 是判断当算法陷入局部最优解时,启动填充函数寻优机制方法选择新采样点,跳出局部最优。

步骤 5.2 是通过式(8)构造在  $x_{\text{new}}$  处的填充函数  $P(x, x^*, p, a)$ 。

步骤 5.3 中,  $x_{\text{new}}$  是填充函数  $p(x, x_{\text{new}})$  的一个局部极大值点,是响应面的平稳点。以  $x_{\text{new}}$  为起点采用局部下降算法,则基于梯度的算法收敛到的解仍为  $x_{\text{new}}$ 。通常填充函数方法会对其执行一个小的扰动产生  $x_0$ ,本文对  $x_{\text{new}}$  的每个维度施加正态分布  $N(0, \sigma^2)$  扰动产生  $x_0$ 。数学表达式如下:

$$x_0 = x_{\text{new}} + [\delta_1, \delta_2, \dots, \delta_d]^T, \delta_i \sim N(0, \sigma^2), i = 1, \dots, d. \quad (11)$$

首次扰动取  $\sigma = 1$ ,即标准正态分布扰动。随着迭代的进行,算法收敛到更优的局部最优解,式(11)中扰动的方差逐渐减小至 0,随迭代步减小的表达式为  $\sigma_n = \frac{N-n+1}{N-n_0}$ ,  $n$  为迭代次数。 $x_0$  与  $x_{\text{new}}$  在  $p(x, x_{\text{new}})$  的同一个峰里,并且  $x_0$  不是平稳点。以  $x_0$  为起点启动局部优化算法收敛到  $x'_{\text{new}}$ ,按照填充函数的构造依据,有  $x'_{\text{new}}$  必然位于  $f(x)$  的一个盆谷里<sup>[12]</sup>。

步骤 5.4 完成了填充函数跳出机制。以  $x'_{\text{new}}$  为起点求解优化问题(9)得到  $x^*_{\text{new}}$ ,即可获得比  $x_{\text{new}}$  更好的局部最优解。

以下是 CORSFFM 算法的流程图。

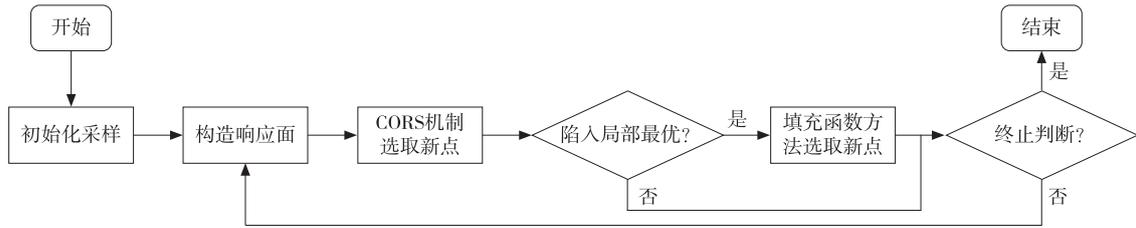


图 1 昂贵黑箱优化问题的响应面算法流程图

Fig. 1 Flow diagram of response surface algorithm for expensive black-box optimization

### 3 数值实验

为了检验本文算法的有效性,在 37 个标准测试问题上展开了数值实验。37 个测试函数均来自于公开的标准数值优化函数库(<http://www.sfu.ca/~ssurjano/index.html>),该函数库拥有大量的针对各种情形的测试函数,比如常见的多极值函数 RASTRIGIN、ACKLEY 等都来自于该函数库。为了进一步验证本文算法的有效性,选取除 CORS 算法之外的 AMGO 算法<sup>[8]</sup>、NRO 算法<sup>[9]</sup>和 Stochastic-RBF 算法(StoRBF 算法)<sup>[14]</sup>这 3 个同类算法进行了对比数值实验。

CORSFFM 算法具有一定的随机性,随机性来源于 3 个方面:一是算法初始化阶段,使用拉丁超立方设计进行随机采样;二是使用 MATLAB 内置 fmincom 函数求解优化问题(5)时需要随机产生初始解;三是启用填充函数寻优机制跳出局部最优时,即求解优化问题(10)时采用了正态分布扰动。本文通过固定随机种子的方式产生相同的初始化采样点以消除算法初始化阶段的随机性,并且对每个测试问题均重复 30 次数值实验,取平均值进行比较,从而减小算法运行结果的随机性。

初始采样点数量  $n_0 = 2(d + 1)$ ,其中  $d$  是测试问题的维度。所有算法的终止准则是最大估值次数  $M = 200$ ,通过调参得到  $T = 15$  算法性能较好。本文所有算法均在 Ubuntu 18.04.6 LTS 平台下执行,计算机处理器型号为 Intel(R) Xeon(R) W-2255CPU @3.70 GHz  $\times$  20,使用的软件环境是 Matlab 2018a。

用  $N$  表示总测试问题数量( $N = 37 \times 30$ ),其中最优解比 CORS 算法 30 计算最优解的平均值更优的测试问题数量为  $T$ ,相对占优百分比定义为  $Q = T/N$ 。表 2 给出各个算法的对于 CORS 算法的相对占优百分比。

表 2 相对占优百分比

Tab. 2 Dominant percentage relative

%

指标	CORSFFM 算法	AMGO 算法	NRO 算法	StoRBF 算法
Q	76.46	56.86	19.60	27.45

表 2 中百分数表示相对于 CORS 算法,  $Q$  表示该算法计算的最优解更优的测试函数占测试函数的百分比。从该表可以看出,CORSFFM 算法在绝大多数测试问题上计算的结果优于 CORS 算法和其他对比算。

表 3 给出更加详细的各算法的最优值计算结果。该表中加粗字体表示该测试算法最优,表中最优值为 30 实验的最优值的平均值。

本文使用文献[23]提出的算法性能表现函数和文献[24]提出的次数表现函数作为指标来综合评价各算法的数值性能。

图 2 的曲线分别代表每种算法的次数表现函数,越在上方的曲线表示对应算法求解到的最优值越好。图 3 是各算法的次数表现函数,图像中曲线越高表示该算法求得的全局最优解在 37 个测试问题当中总体越好。由图 2 和图 3 可以看出在两种指标上,代表 CORSFFM 算法的曲线均高于其他曲线,这表明,本文 CORSFFM 算法优于 CORS 算法和其他算法。

图 4 到图 9 表示算法求得的最优解随着迭代次数的下降曲线。为消除初始化采样给算法带来的影响,上图中的曲线表示对该测试函数 30 次独立实验的平均下降趋势。从以上图像中可以看出,CORSFFM 算法能够在较少的估值次数的情形下收敛的较好的近似最优解。

表 3 各对比算法最小值比较

Tab. 3 Comparison of minimal values of algorithms

测试问题	CORSFFM 算法	AMGO 算法	CORS 算法	NRO 算法	StoRBF 算法
Ackley_30	<b>5.49E+00</b>	7.20E+00	1.14E+01	2.02E+01	1.54E+01
Branin	<b>3.98E-01</b>	3.98E-01	3.98E-01	9.55E-01	3.98E-01
Colville_4	<b>3.52E+01</b>	7.38E+01	2.36E+02	2.47E+02	3.54E+01
Perm_2	<b>5.63E-08</b>	1.32E-04	4.94E-07	3.35E-05	2.20E-03
Powell_2	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
Powell_4	1.28E+00	1.72E+00	9.50E+00	4.37E+00	1.12E-01
Styblinski_10	<b>-3.50E+02</b>	-3.38E+02	-2.95E+02	-3.18E+02	-3.42E+02
Styblinski_2	<b>-7.83E+01</b>	<b>-7.83E+01</b>	<b>-7.83E+01</b>	-7.70E+01	-7.55E+01
beale_2	4.27E-02	1.36E-01	6.78E-02	2.35E-01	<b>4.56E-03</b>
boha_1	7.29E-02	8.67E-02	<b>2.77E-02</b>	2.78E-01	2.82E-01
boha_2	1.47E-02	5.84E-02	<b>9.21E-03</b>	2.47E-02	1.83E-01
boha_3	4.17E-02	7.40E-02	1.59E-02	<b>4.05E-02</b>	1.38E-01
booth	<b>1.71E-12</b>	4.41E-10	2.29E-11	3.07E-03	6.27E-04
bukin	<b>1.52E+00</b>	1.95E+00	3.10E+00	8.59E+00	1.67E+00
camel3	<b>9.92E-09</b>	1.52E-05	1.90E-06	9.35E-02	4.53E-05
camel6	-1.03E+00	-1.03E+00	-1.03E+00	-1.01E+00	<b>-9.23E-01</b>
crossit	<b>-2.06E+00</b>	<b>-2.06E+00</b>	<b>-2.06E+00</b>	-2.04E+00	-1.95E+00
dixon_2	7.30E-05	<b>3.44E-05</b>	1.64E-03	3.60E-01	7.14E-04
dixon_4	4.56E+00	7.83E+00	3.84E+01	9.29E+00	<b>3.66E-01</b>
dixon_6	3.04E+01	2.85E+01	1.28E+02	4.90E+01	<b>7.13E-01</b>
drop	-8.45E-01	-8.24E-01	-8.32E-01	<b>-8.77E-01</b>	-7.95E-01
easom_2	-1.73E-02	-2.45E-02	<b>-5.57E-02</b>	-1.86E-01	-3.54E-01
egg	<b>-9.04E+02</b>	-8.75E+02	-8.88E+02	-8.07E+02	-6.48E+02
goldsteinPrice	<b>3.05E+00</b>	5.16E+00	3.22E+00	6.31E+00	1.02E+01
goldsteinPriceScaled	<b>-3.13E+00</b>	<b>-3.13E+00</b>	<b>-3.13E+00</b>	-2.84E+00	-2.81E+00
grlee12	<b>-1.25E+00</b>	<b>-1.25E+00</b>	<b>-1.25E+00</b>	<b>-1.25E+00</b>	<b>-1.25E+00</b>
hartman_3	<b>-3.86E+00</b>	<b>-3.86E+00</b>	<b>-3.86E+00</b>	-3.74E+00	-3.75E+00
hartman_4	<b>-3.13E+00</b>	<b>-3.13E+00</b>	<b>-3.13E+00</b>	-2.87E+00	-3.06E+00
hartman_6	-3.02E+00	<b>-3.04E+00</b>	<b>-3.04E+00</b>	-2.70E+00	-3.01E+00
levy13	<b>2.73E-02</b>	1.39E-01	1.05E-01	2.13E-01	6.48E-02
matyas	<b>4.86E-13</b>	4.83E-12	7.45E-12	2.85E-03	2.19E-05
rastrign_2	<b>8.76E-01</b>	1.66E+00	1.94E+00	3.19E+00	4.02E+00
rastrign_6	<b>2.33E+01</b>	2.59E+01	3.16E+01	2.43E+01	2.80E+01
rosenbrock_2	<b>8.15E-04</b>	2.81E-03	2.62E-03	1.59E-01	6.93E-02
rosenbrock_4	1.45E+00	1.59E+00	<b>1.38E+00</b>	2.04E+00	1.83E+00
rosenbrock_6	4.47E+00	4.53E+00	1.04E+01	8.47E+00	<b>3.99E+00</b>
shekel	<b>-7.83E+00</b>	-5.49E+00	-4.00E+00	-2.97E+00	-4.14E+00

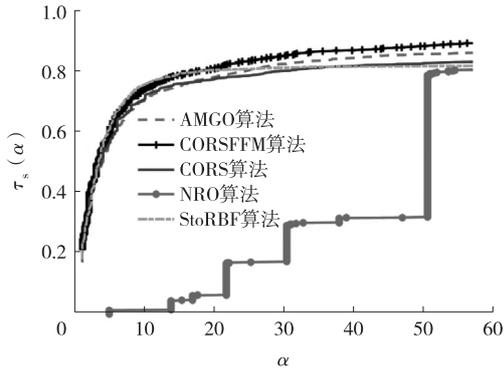


图 2 次数表现曲线

Fig. 2 Numerical of performance curve

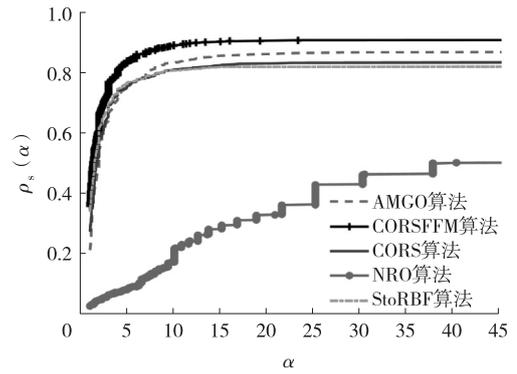


图 3 性能表现曲线

Fig. 3 Performance curve

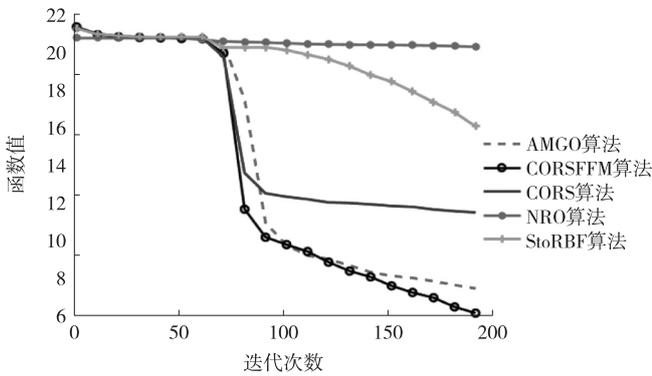


图 4 Ackley\_30

Fig. 4 Ackley\_30

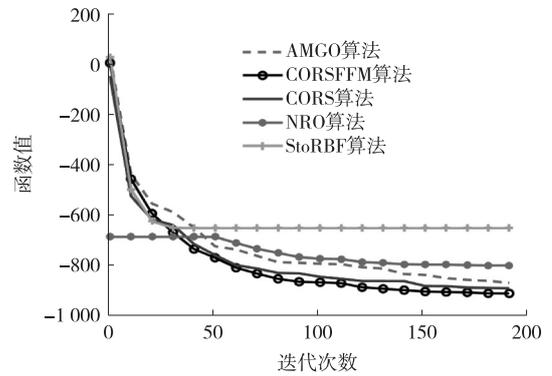


图 5 egg

Fig. 5 egg

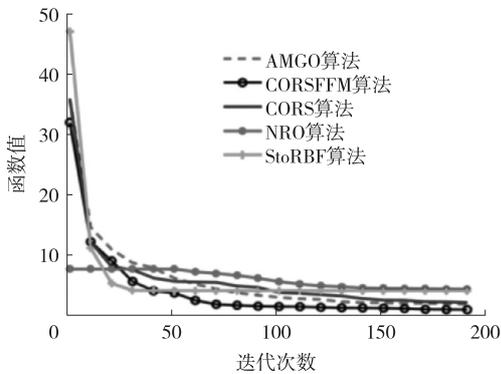


图 6 rastrign\_2

Fig. 6 rastrign\_2

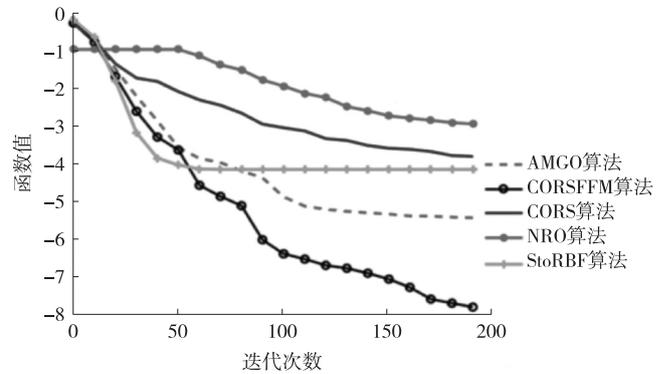


图 7 shekel

Fig. 7 shekel

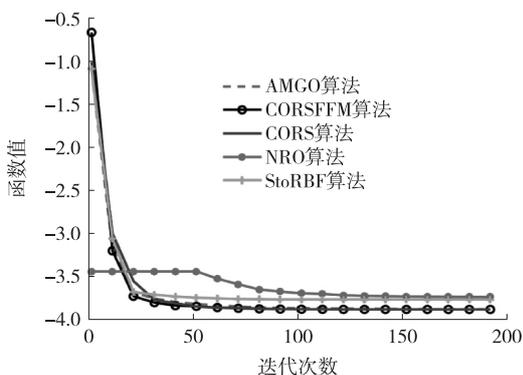


图 8 hartman\_3

Fig. 8 hartman\_3

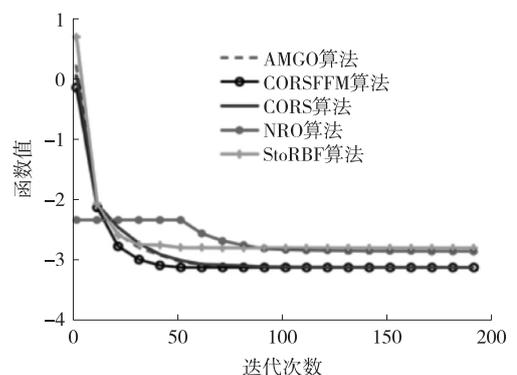


图 9 goldsteinPriceScaled

Fig. 9 goldsteinPriceScaled

## 4 结束语

本文从 CORS 算法出发,通过引入填充函数寻优机制辅助产生新采样点,研究了昂贵黑箱函数优化问题,提出了 CORSFFM 算法。该算法具有自适应的平衡局部搜索和全局搜索机制,利用填充函数方法跳出局部最优解。通过数值实验验证了算法的数值性能优于原算法。本文工作为研究平衡昂贵黑箱优化局部搜索和全局搜索机制提供了新的思路。需要指出的是 CORSFFM 的性能依赖于参数  $T$  的选取,如何根据优化问题的特殊性更准确地刻画  $T$  将是进一步研究的重点。

### 参考文献:

- [1] MÜLLER J, PARK J, SAHU R, et al. Surrogate optimization of deep neural networks for groundwater predictions[J]. *Journal of Global Optimization*, 2021, 81(1): 203-231.
- [2] AKHTAR T, SHOEMAKER C A. Multi objective optimization of computationally expensive multi-modal functions with RBF surrogates and multi-rule selection[J]. *Journal of Global Optimization*, 2016, 64(1): 17-32.
- [3] TANG Y, CHEN J, WEI J, et al. A surrogate-based particle swarm optimization algorithm for solving optimization problems with expensive black box functions[J]. *Engineering Optimization*, 2013, 45(5): 557-576.
- [4] AUDET C, DIGABEL S L, TRIBES C, et al. Dynamic scaling in the mesh adaptive direct search algorithm for blackbox optimization[J]. *Optimization & Engineering*, 2016, 17(2): 333-358.
- [5] JONES D R, SCHONLAU M W, William J, et al. Efficient global optimization of expensive black-box functions[J]. *Journal of Global Optimization*, 1998, 13(4): 455-492.
- [6] GUTMANN H M. A radial basis function method for global optimization[J]. *Journal of Global Optimization*, 2001, 19(3): 201-227.
- [7] REGIS R G, SHOEMAKER C A. Constrained global optimization of expensive black box functions using radial basis functions [J]. *Journal of Global Optimization*, 2005, 31(1): 153-171.
- [8] JIE H X, WU Y Z, DING J W, et al. An adaptive metamodel-based global optimization algorithm for black-box type problems [J]. *Engineering Optimization*, 2015, 47(11): 1-22.
- [9] ZHOU Y, HE X, CHEN Z, et al. A neighborhood regression optimization algorithm for computationally expensive optimization problems[J]. *Institute of Electrical and Electronics Engineers Transactions on Cybernetics*, 2022, 52(5): 3018-3031.
- [10] GE R. A filled function method for finding a global minimizer of a function of several variables[J]. *Mathematical Programming*, 1990, 46(3): 191-204.
- [11] ZHU W X. A class of filled functions for box constrained continuous global optimization [J]. *Applied Mathematics & Computation*, 2005, 169(1): 129-145.
- [12] LIU X. A class of generalized filled functions with improved computability [J]. *Journal of Computational and Applied Mathematics*, 2001, 137(1): 61-69.
- [13] POWELL M J D. UOBYQA: unconstrained optimization by quadratic approximation[J]. *Mathematical Programming*, 2002, 92(3): 555-582.
- [14] REGIS R G, SHOEMAKER C A. A stochastic radial basis function method for the global optimization of expensive functions [J]. *European Journal of Operational Research*, 2007, 19(4): 497-509.
- [15] ZHOU Z, BAI F S. An adaptive framework for costly black-box global optimization based on radial basis function interpolation [J]. *Journal of Global Optimization*, 2018, 70(4): 757-781.
- [16] LI G, ZHANG Q F, LIN Q, et al. A three-level radial basis function method for expensive optimization[J]. *Institute of Electrical and Electronics Engineers Transactions on Cybernetics*, 2021, 52(7): 1-12.
- [17] AKBARI H, KAZEROONI A. KASRA: a kriging-based adaptive space reduction algorithm for global optimization of computationally expensive black-box constrained problems[J]. *Applied Soft Computing*, 2020, 90(2): 106-154.
- [18] CLARKE S M, GRIEBSCH J H, SIMPSON T W, et al. Analysis of support vector regression for approximation of complex engineering analyses[J]. *Journal of Mechanical Design*, 2005, 127(6): 1077-1087.
- [19] YUN Y, YOON M, NAKAYAMA H, et al. Multi-objective optimization based on meta-modeling by using support vector regression[J]. *Optimization & Engineering*, 2009, 10(2): 167-181.
- [20] GOEL T, HAFTKA R T, SHYY W, et al. Ensemble of surrogates[J]. *Structural and Multidisciplinary Optimization*, 2007, 33

(3);199-216.

- [21] 陈文,傅卓佳,魏星.科学与工程计算中的径向基函数方法[M].北京:科学出版社,2014.  
CHEN W,FU Z J,WEI X. Radial basis function method in scientific and engineering computing[M]. Beijing: Science Press, 2014.
- [22] MCKAY M D,BECKMAN R J,CONOVER W J,et al. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code[J]. Echnometrics,1979,42(1):55-61.
- [23] DOLAN E D, MORE J J. Benchmarking optimization software with performance profiles[J]. Mathematical Programming, 2002,91(2):201-213.
- [24] MORÉ J J, WILD S M. Benchmarking derivative-free optimization algorithms[J]. SIAM Journal on Optimization,2009,20(1): 172-191.

## Operations Research and Cybernetics

### An Adaptive Response Surface Method Based on Escape Local Optimal Solution Mechanism

SHEN Fuwei<sup>1</sup>, BAI Fusheng<sup>2</sup>

(1. School of Mathematical Sciences, Chongqing Normal University ;

2. National Center for Applied Mathematics in Chongqing, Chongqing 401331, China)

**Abstract:** [Purposes]A response surface method is proposed for the optimization of expensive black box functions, which can better balance local search and global search. [Methods]The CORS method is improved, introducing the mechanism of judging the local optimum solution to CORS, then jump out of it using filled function method. [Findings]Numerical experiments on 37 open global optimization algorithm test problems show that the adaptive response surface algorithm based on the local jump out local optimal mechanism of filled functions can converge to a better approximate global optimal solution. [Conclusions]The local optimal solution jumping mechanism based on the filled function method can improve the global search ability of the algorithm.

**Keywords:** filled function; expensive black-box function; response method; radial basis function

(责任编辑 陈 乔)