

## 基于约束优化的混合径向基函数响应面方法<sup>\*</sup>

殷悦, 白富生

(重庆师范大学 数学科学学院, 重庆 401331)

**摘要:**【目的】为更好地求解昂贵黑箱函数优化问题。【方法】在响应面约束优化算法的基础上,采用由薄板样条径向基函数模型和三次径向基函数模型的组合得到的混合模型作为新响应面模型对黑箱函数进行逼近,并且在优化过程中自适应选择混合模型的权重因子。【结果】得到了混合径向基函数响应面算法。【结论】在 34 个常用检验问题上得到的算法数值实验结果表明所提出的算法优于原有的响应面约束优化算法、OPUS 算法和 SAMSO 算法。

**关键词:**昂贵黑箱函数;全局优化;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(2021)01-0022-08

在工程、科学、金融、管理等诸多领域中常常涉及到昂贵黑箱函数,该函数的特点在于函数解析表达式未知、函数估值耗时或需要花费大量资源等。求解昂贵黑箱函数最优点和最优值的研究具有重要的实践意义,目前在求解昂贵黑箱优化问题的理论与方法上已经取得了一系列的成果<sup>[1-7]</sup>。下面给出问题的数学模型,即:

$$\begin{aligned} \min & f(x) \\ \text{s. t. } & a \leq x \leq b, x \in \mathbf{R}^d \end{aligned} \quad (1)$$

其中:  $f(x): \mathbf{R}^d \rightarrow \mathbf{R}$ , 是一个表达式未知且计算耗时的昂贵黑箱函数,  $\mathcal{D} = [a, b] \in \mathbf{R}^d$ 。全局优化算法旨在可行域  $\mathcal{D}$  中找到一个  $x^*$ , 使得  $\forall x \in \mathcal{D}$  都有  $f(x^*) \leq f(x)$ 。解决问题(1)的常用方法之一是响应面方法,也称之为代理模型或元模型方法<sup>[1]</sup>。常见的响应面模型包括低次多项式模型<sup>[2]</sup>、克里金(Kriging)模型<sup>[3]</sup>及径向基函数(Radial basis function)模型<sup>[4]</sup>。径向基函数具有形式简单、系数易确定、可应用于高维问题等特点。

2001 年, Gutmann<sup>[4]</sup>提出了一种基于径向基响应面的全局优化算法(Gutmann-RBF),该算法将使得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>[5]</sup>提出了响应面约束优化(Constrained optimization using response surfaces, CORS)算法,该算法在求解以响应面模型为目标函数的辅助问题中增加距离约束以控制下一个采样点与已有采样点的距离,数值试验表明 CORS 算法在多数情况下算法效率优于 Gutmann-RBF 算法。2007 年, Regis 等人<sup>[6]</sup>又依据距离准则和响应面函数值准则,在每次迭代随机产生的多个候选点中选出一个候选点作为新采样点。2010 年, Acar<sup>[7]</sup>提出响应面模型的拟合精度可以通过集成单个响应面模型来提高。2014 年, Regis<sup>[8]</sup>将粒子群算法与径向基响应面结合提出了使用替代函数的粒子群优化(Optimization by particle swarm using surrogates, OPUS)算法,该算法在每次迭代的过程中利用响应面模型对粒子群算法中的粒子进行筛选,以便快速获得更优质粒子的信息。2015 年, Jie 等人<sup>[9]</sup>提出了 AMGO 算法,在算法中使用了克里金模型和三次径向基函数模型的组合作为昂贵黑箱函数的响应面模型。2017 年, Müller<sup>[10]</sup>提出了求解昂贵多目标问题的优化算法 SOCEMO,对多个目标同时构建响应面模型,并利用响应面模型的信息选择新的候选点集。2018 年, Zhou 等人<sup>[11]</sup>在 Gutmann-RBF 算法的基础上增加了比较连续两次找到的响应面模型最优点之间距离的步骤:如距离很近,则说明当前的径向基响应面模型的全局最优点可能在原黑箱目标函数的局部最优点或全局最优点附近,故将当前响应面的最优点视为新采样点;此算法取得了良好的数值试验结果。2019 年, Müller 等人<sup>[12]</sup>提出了一种解决具有隐藏约束的黑箱问题的优化算法 SHEBO,该算法通过循环执行局

\* 收稿日期:2020-11-17 修回日期:2021-01-06 网络出版时间:2021-03-11 16:37

资助项目:国家自然科学基金(No. 11871128);重庆市自然科学基金(No. cstc2019jcyj-msxmX0368;No. cstc2018jcyjAX0172)

第一作者简介:殷悦,女,研究方向为最优化理论与算法,E-mail:y1303349462@163.com;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.20210310.1418.020.html

部搜索、转换搜索和全局搜索的来寻找下一个候选点。2020 年, Li 等人<sup>[13]</sup>提出了一种求解昂贵黑箱优化问题的响应面模型辅助的双种群智能优化算法 SAMSO。

通过采用文献[9]中类似的函数模型组合方法, 本文对 CORS 算法进行了改进, 提出了混合响应面约束优化算法。在 CORS 算法中, 只采用三次径向基函数作为响应面模型。本文采用三次径向基函数和薄板样条径向基函数的组合作为响应面模型, 在迭代过程中根据两种响应面模型的函数值信息自适应调整两种模型的权重因子, 并在 34 个标准检验函数上进行了数值实验。计算结果表明与 CORS 算法、OPUS 算法及 SAMSO 算法相比, 本文提出的基于混合径向基函数响应面的算法具有明显的优势。

## 1 预备知识

本文的方法基于求解全局优化问题的径向基函数方法, 故首先对径向基函数响应面模型做一简略的介绍。

### 1.1 径向基函数响应面模型

设  $x_1, x_2, \dots, x_n \in \mathbf{R}^d$  为已知函数值  $f(x_1), f(x_2), \dots, f(x_n)$  的采样点, 基于数据集  $\{(x_1, f(x_1)), (x_2, f(x_2)), \dots, (x_n, f(x_n))\}$  构造径向基函数模型, 该模型可表示为:

$$S_n(x) = \sum_{i=1}^n \lambda_i \varphi(\|x - x_i\|) + p(x), x \in \mathbf{R}^d. \quad (2)$$

其中:  $f$  是一个确定性的连续函数,  $\|\cdot\|$  是定义在  $\mathbf{R}^d$  上的欧氏范数,  $\lambda_1, \lambda_2, \dots, \lambda_n \in \mathbf{R}^d$ 。  $\varphi$  为径向基函数, 表达式如表 1 所示,  $p \in \Pi_m^d$  ( $\Pi_m^d$  表示次数不超过  $m$  的多项式组成的空间)。

表 1 不同形式的径向基函数

Tab. 1 Radial basis functions in different forms

| 径向基函数                    | $\varphi(r)$                        | $m_\varphi$ | $p(x)$ (当 $m=m_\varphi$ 时) |
|--------------------------|-------------------------------------|-------------|----------------------------|
| 三次 (Cubic)               | $r^3$                               | 1           | $\mathbf{b}^T x + a$       |
| 薄板样条 (Thin plate spline) | $r^2 \ln r$                         | 1           | $\mathbf{b}^T x + a$       |
| 线性 (Linear)              | $r$                                 | 0           | $a$                        |
| 多二次 (Multiquadric)       | $\sqrt{r^2 + \gamma^2}, \gamma > 0$ | 0           | $a$                        |
| 高斯 (Gaussian)            | $\exp(-\gamma r^2), \gamma > 0$     | -1          | $\{0\}$                    |

设  $\hat{m}$  为线性空间  $\Pi_m^d$  的基的个数,  $m_\varphi$  表示相应多项式可选的最小次数<sup>[1]</sup>, 令  $p_1, p_2, \dots, p_{\hat{m}}$  为  $\Pi_m^d$  的一组基, 则  $p$  可由  $p_1, p_2, \dots, p_{\hat{m}}$  的线性组合表出。定义  $n \times \hat{m}$  矩阵

$$\mathbf{P} = \begin{pmatrix} p_1(x_1) & p_2(x_1) & \cdots & p_{\hat{m}}(x_1) \\ p_1(x_2) & p_2(x_2) & \cdots & p_{\hat{m}}(x_2) \\ \vdots & \vdots & \vdots & \vdots \\ p_1(x_n) & p_2(x_n) & \cdots & p_{\hat{m}}(x_n) \end{pmatrix}$$

以及

$$\Phi = \begin{pmatrix} \varphi(\|x_1 - x_1\|) & \varphi(\|x_1 - x_2\|) & \cdots & \varphi(\|x_1 - x_n\|) \\ \varphi(\|x_2 - x_1\|) & \varphi(\|x_2 - x_2\|) & \cdots & \varphi(\|x_2 - x_n\|) \\ \vdots & \vdots & \vdots & \vdots \\ \varphi(\|x_n - x_1\|) & \varphi(\|x_n - x_2\|) & \cdots & \varphi(\|x_n - x_n\|) \end{pmatrix},$$

由文献[6]可知,  $(n + \hat{m}) \times (n + \hat{m})$  矩阵  $\mathbf{A} = \begin{pmatrix} \Phi & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{pmatrix}$  是非奇异的当且仅当  $x_1, x_2, \dots, x_n$  满足:  $q \in \Pi_m^d$  且  $q(x_i) = 0$ ,

$i = 1, \dots, n$  蕴含  $q \equiv 0$ , 此时可通过求解线性方程组  $\mathbf{A} \begin{pmatrix} \lambda \\ c \end{pmatrix} = \begin{pmatrix} \mathbf{F} \\ \mathbf{0}_{\hat{m}} \end{pmatrix}$ , 唯一确定径向基函数模型(2)的系数, 其中  $\lambda =$

$(\lambda_1, \lambda_2, \dots, \lambda_n)^T, c \in \mathbf{R}^{\hat{m}}, c$  中的元素是多项式  $p$  由基  $p_1, p_2, \dots, p_{\hat{m}}$  线性表出时的相应系数,  $\mathbf{0}_{\hat{m}}$  是  $\mathbf{R}^{\hat{m}}$  中的零向量。

当  $p(x)$  为线性函数时,如果矩阵  $\mathbf{P}$  的秩为  $d+1$ ,即列满秩,则矩阵  $\mathbf{A}$  非奇异,故此时(2)式的系数可唯一确定。

## 1.2 CORS 方法

响应面约束优化(CORS)算法中,初始点的选取为可行域的顶点。新的采样点通过解决如下辅助问题来获得,有:

$$\begin{aligned} \min & \hat{f}_n(x), \\ \text{s. t. } & \|x - x_j\| \geq \beta_n \Delta_n, j=1, \dots, n, x \in \mathcal{D}. \end{aligned} \quad (3)$$

其中:  $\hat{f}_n(x)$  为当前响应面模型,  $x_1, x_2, \dots, x_n$  为已有采样点,  $\Delta_n = \max_{x \in \mathcal{D}} \min_{1 \leq j \leq n} \|x - x_j\|$ ,  $\beta_n \in [0, 1)$ 。  $\beta_n$  的选取方式为:给定包含  $M$  个元素的搜索模式向量  $\langle \gamma_1, \gamma_2, \dots, \gamma_M \rangle$ ,  $\gamma_1, \gamma_2, \dots, \gamma_M$  逐次递减且  $\gamma_M = 0$ , 令

$$\beta_n = \begin{cases} \gamma_{\text{mod}(n-n_0+1, M)}, & \text{mod}(n-n_0+1, M) \neq 0 \\ \gamma_M, & \text{mod}(n-n_0+1, M) = 0 \end{cases},$$

其中:  $n \geq n_0$ ,  $n_0$  是初始采样点的个数。  $\beta_n$  用于控制候选点与已采样点的位置关系,当  $\beta_n$  取值较大时,(3)式中的约束条件要求候选点的位置远离已采样点,此时意味着做全局搜索;当  $\beta_n$  较小时,候选点的位置接近已采样点,此时意味着做局部搜索。当  $\beta_n = 0$  时,约束条件不起作用,此时相当于求解一个无约束问题。如果(3)式中的响应面模型  $\hat{f}_n(x)$  采用径向基插值函数  $s_n(x)$ , 则相应响应面约束优化方法称为径向基函数响应面约束优化(CORS-RBF)算法<sup>[5]</sup>。

## 2 混合响应面约束优化算法

如何建立一个响应面模型是提高全局优化算法效率和模型拟合精度的关键。CORS 算法采用三次径向基函数建立响应面模型,这种选取方式建立的模型精度有限,所以考虑采用两种径向基函数模型的组合模型,两种径向基函数模型分别选择三次和薄板样条径向基函数。模型建立方式如下:

$$u_n(x) = w_n^t s_{n-1}^t(x) + w_n^c s_{n-1}^c(x), w_n^t + w_n^c = 1.$$

其中:  $n \geq n_0$ ,  $s_n^t(x)$  和  $s_n^c(x)$  分别表示采用薄板样条径向基函数和三次函数构建的响应面模型,  $w_n^t, w_n^c$  分别是这两种模型的权重因子,计算方式为:

$$w_n^c = \frac{e_n^t}{e_n^t + e_n^c}, w_n^t = 1 - w_n^c. \quad (4)$$

其中:  $e_n^t = |f(x_n) - s_{n-1}^t(x_n)|$ ,  $e_n^c = |f(x_n) - s_{n-1}^c(x_n)|$ ; 当  $n = n_0$  时,  $w_{n_0}^c = 0.5$ ,  $w_{n_0}^t = 0.5$ ;  $x_{n-1}$  为第  $n-1$  个昂贵估值点,  $f(x_{n-1})$  为  $x_{n-1}$  的真实函数值,  $s_{n-1}^t(x_{n-1})$ ,  $s_{n-1}^c(x_{n-1})$  分别为两种径向基函数模型在  $x_{n-1}$  处的函数值。

下面给出完整的混合响应面约束优化算法流程。

输入:

- 1) 一个定义在闭的超矩形  $\mathcal{D} = [a, b] \in \mathbf{R}^d$  上的目标函数  $f$ ;
- 2) 响应面模型  $u_n(x), s_n^t(x), s_n^c(x)$ ;
- 3) 通过初始点实验设计生成的初始取样点集  $\mathcal{A}_{n_0} = \{x_1, x_2, \dots, x_{n_0}\} \subset \mathcal{D}$ ;
- 4) 一个  $M$  维搜索模式向量  $\langle \gamma_1, \gamma_2, \dots, \gamma_M \rangle$ , 其中  $\gamma_j \in [0, 1)$  关于  $j = 1, 2, \dots, M$  递减;
- 5)  $w_{n_0}^c = 0.5, w_{n_0}^t = 0.5$ ;
- 6) 允许进行原目标函数计算的最大次数  $N_{\max}$  (默认值:  $N_{\max} = 200$ )。

输出: 算法结束之后所计算出的最好取样点  $x_{N_{\max}}^*$ , 以及对应的目标函数值  $f_{N_{\max}}^*$ 。

第 1 步, 初始化。计算初始采样点的目标函数值  $f(x_1), f(x_2), \dots, f(x_{n_0})$ , 令  $n = n_0$ , 数据集  $\mathcal{B}_n = \{(x_1, f(x_1)), \dots, (x_n, f(x_n))\}$ 。

第 2 步, 找到当前最好的点。找出  $x_n^* = \text{argmin}\{f(x) : x \in \mathcal{A}_n\}$ , 令  $f_n^* = f(x_n^*)$ 。

第 3 步, 迭代。当终止条件不满足(即当  $n \leq N_{\max}$ )时:

1) 更新响应面模型的权重因子。使用数据集  $\mathcal{B}_n$  建立/更新响应面模型  $s_n^t(x), s_n^c(x)$ , 根据(4)式计算权重因子  $w_n^c, w_n^t$ 。

2) 利用已有数据建立/更新响应面模型。使用权重因子  $w_c, w_t$  建立/更新响应面模型  $u_n(x)$ , 计算响应面模

型的全局最优值点  $z_n^* = \operatorname{argmin}\{u_n(x), x \in \mathcal{D}\}$ 。

3) 选取新的采样点。求解全局优化问题:

$$\begin{aligned} \min u_n(x), \\ \text{s. t. } \|x - x_j\| \geq \beta_n \Delta_n, x_j \in \mathcal{A}_n. \end{aligned} \quad (5)$$

得到全局最优值点作为下一个迭代点  $x_{n+1}$ 。

4) 做昂贵函数估值。计算候选点  $x_{n+1}$  的目标函数值。

5) 更新当前最好的点及目标函数值。如果  $f(x_{n+1}) < f_{\text{best}}^n$ , 那么令  $x_{n+1}^* = x_{n+1}, f_n^* = f(x_{n+1})$ ; 否则令  $x_{n+1}^* := x_n^*, f_{n+1}^* := f_n^*$ 。

第 4 步, 返回算法结果。返回找到的近似全局极小点  $x_{N_{\max}}^*$  及相应目标函数值  $f_{N_{\max}}^*$ 。

在输入中, 目标函数必须是确定性的, 即对同一个输入的自变量, 输出的目标函数值是唯一确定的, 搜索空间是一个闭的超矩形(输入 1)。响应面模型(输入 2)可以有多种选择。初始采样点集通常采用对称拉丁超立方实验设计生成或由超矩形的顶点组成。在数值实验中, 本文选取由(2)式定义的径向基插值函数  $s_n^t(x), s_n^c(x)$  的组合模型  $u_n(x)$  作为响应面模型。初始点(输入 3)可以在整个搜索空间中随机生成, 算法的表现一定程度上与初始点的在  $\mathcal{D}$  中的分布有关。本文中初始采样点通过对称拉丁超立方体试验设计(SLHD)<sup>[14]</sup>产生,  $n_0 = 2(d+1)$ ,  $d$  表示问题的维度。输入中包括了搜索模式向量的取值(输入 4)和权重因子  $w_n^t, w_n^c$  的初始值(输入 5), 随着迭代的进行  $w_n^t, w_n^c$  的计算由(4)式给出, 在实验中若出现  $e_n^t + e_n^c$  的值接近 0 则重置的  $w_n^t, w_n^c$  值为 0.5。由于目标函数值的估值昂贵性, 本文设定允许进行目标函数估值的最大次数  $N_{\max}$ 。改进后的算法流程图如下。

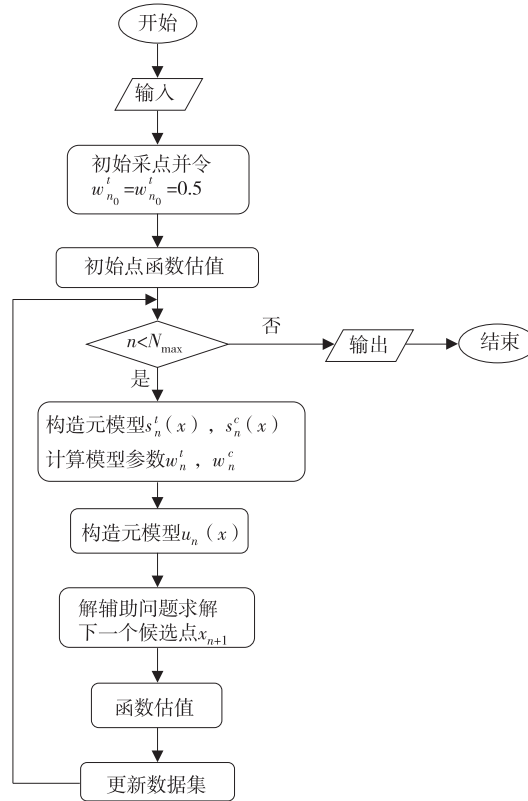


图 1 混合响应面约束优化算法流程

Fig. 1 The algorithm flow of hybrid response surface constrained optimization

### 3 数值试验

用于对比 CORS 算法、混合响应面约束优化算法、OPUS 算法和 SAMSO 算法的 34 个测试问题维数在 2 到 30 之间, 其中包含 7 个 Dixon-Szegö<sup>[15]</sup> 测试问题 (Branin, Goldstein-Price, Hartman3, Shekel5, Shekel7, Shekel10, Hartman6) 和 2 个具有大量局部极小点的测试问题 (Rastrigin, Shubert)<sup>[16]</sup>, 以及 10 个具有相同表达

形式,但维度不同的 Schoen 测试问题<sup>[7,17]</sup>,此外还有 15 个全局优化测试函数库中的测试函数<sup>[18]</sup>。为了对比不同算法在测试函数集上的总体表现,引入性能表现函数  $\rho_s(\alpha)$ <sup>[19]</sup> 和次数表现函数  $\tau_s(\alpha)$ <sup>[20]</sup>。算法的性能表现函数和次数表现函数值越大,算法越有效。

### 3.1 性能表现函数与次数表现函数

为了对比不同算法在测试函数集上的总体表现,下面引入性能表现函数<sup>[19]</sup>和次数表现函数<sup>[20]</sup>的定义。设  $\mathcal{S}$  为要评估的算法的集合,  $\mathcal{P}$  为测试问题的集合。首先给出收敛性测试<sup>[20]</sup>的概念,即:给定一个误差系数  $\delta > 0$ ,在给定目标函数估值次数下,由所有  $\mathcal{S}$  中算法求出的问题  $p$  的最小的最优值  $f_L$ ,如果算法  $s$  在进行了  $k$  次目标函数估值后得到的解  $\bar{x}$  满足:

$$f(x^{(0)}) - f(\bar{x}) \geq (1 - \delta)(f(x^{(0)}) - f_L), \quad (6)$$

则称算法  $s$  此时通过了问题  $p$  的收敛性测试,其中  $x^{(0)}$  表示算法  $s$  使用的初始点。于是,算法  $s(s \in \mathcal{S})$  在问题  $p(p \in \mathcal{P})$  上的性能比率可定义为:  $r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s}: s \in \mathcal{S}\}}$ , 其中  $t_{p,s}$  表示算法  $s$  在问题  $p$  上通过收敛性测试时所需的最少目标函数估值次数。显然  $r_{p,s} \geq 1$ , 并且  $r_{p,s} = 1$  表示在测试问题  $p$  上算法  $s$  的表现最好。

基于性能比率,算法  $s \in \mathcal{S}$  在测试问题集  $\mathcal{P}$  上的性能表现函数  $\rho_s(\alpha)$  定义为<sup>[19]</sup>:  $\rho_s(\alpha) = \frac{1}{|\mathcal{P}|} |\{p \in \mathcal{P}: r_{p,s} \leq \alpha\}|$ , 其中:  $\alpha \geq 1$ ,  $|\cdot|$  表示集合中元素个数。注意到性能表现函数可看作是比率  $r_{p,s}$  的概率分布。由  $\rho_s(\alpha)$  定义可知,算法的性能表现函数值越大,说明算法的效果越好。

对  $s \in \mathcal{S}$ , 次数表现函数  $\tau_s(\alpha)$  的定义为<sup>[19]</sup>:  $\tau_s(\alpha) = \frac{1}{|\mathcal{P}|} |\{p \in \mathcal{P}: \frac{t_{p,s}}{n_p + 1} \leq \alpha\}|$ , 其中:  $n_p$  表示测试问题  $p$  的维数。注意到为了体现出问题维度对达到收敛性测试所需要的目标函数值估值次数的影响,上述定义中采用  $n_p + 1$  作为分母。次数表现函数给出一个算法在一定的目标函数估值次数下能够求解的问题所占比列。算法的次数表现函数值越大,说明该算法越有效。

### 3.2 结果与讨论

这一部分给出了 4 种算法求解所有测试问题的计算结果。4 种算法的程序都在 Windows 环境下的 Matlab 2018a 版本下运行。与文献[5]一致,算法的搜索模式向量选择为  $(0.95, 0.25, 0.05, 0.03, 0)$ 。在求解(5)式时,通过运行 4 次带随机选取初始点 Matlab 优化工具箱中的 fmincon 函数的方式进行求解,(5)式中的  $\Delta_n$  也通过运行 fmincon 函数得到。

在数值实验中,对所有 34 个测试问题,4 种算法各运行 30 次,每次运行的目标函数估值次数上限是 200 次。图 2 给出了 4 种算法的性能表现函数曲线和次数表现函数曲线,曲线越高说明算法的效果越好。对给定的问题  $p$ , (6)式中的  $f_L$  是由每个算法运行 30 次,共 60 次运行得到的最好的最优值,  $\delta$  取 0.05。由于每个算法在每个问题上运行 30 次,计算算法性能表现函数和次数表现函数图像时相当于一共考虑了  $30 \times 34 = 1\,020$  个问题。从图 2 可以看出,混合响应面约束优化算法效果更好。

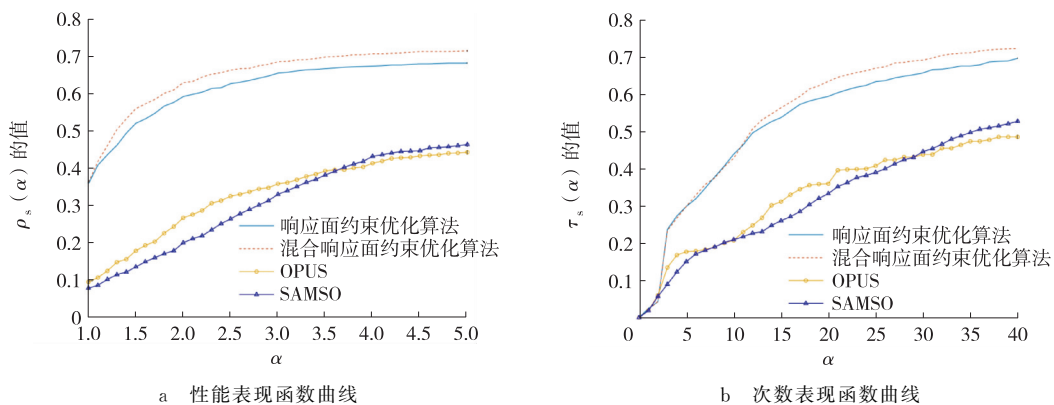


图 2 算法的性能表现函数曲线和次数表现函数曲线

Fig. 2 Performance curve and number performance curve of the algorithm

图 3 给出了 4 种算法求解测试问题 Schoen10\_4\_3, Schoen3Y, Shekel10 和 Shubert 时的优化进程曲线,其中横坐标表示目标函数估值的次数,纵坐标表示在相应目标函数估值次数下算法找到的最优值。考虑到算法具有一定的随机性,每个算法在每个测试问题上都运行 30 次,用平均值作为进程曲线的数据。从图中可以看出当横坐标(函数目标函数值估值次数)达到一定次数后,混合响应面约束优化算法得到的最优函数值比原算法得到的要好。

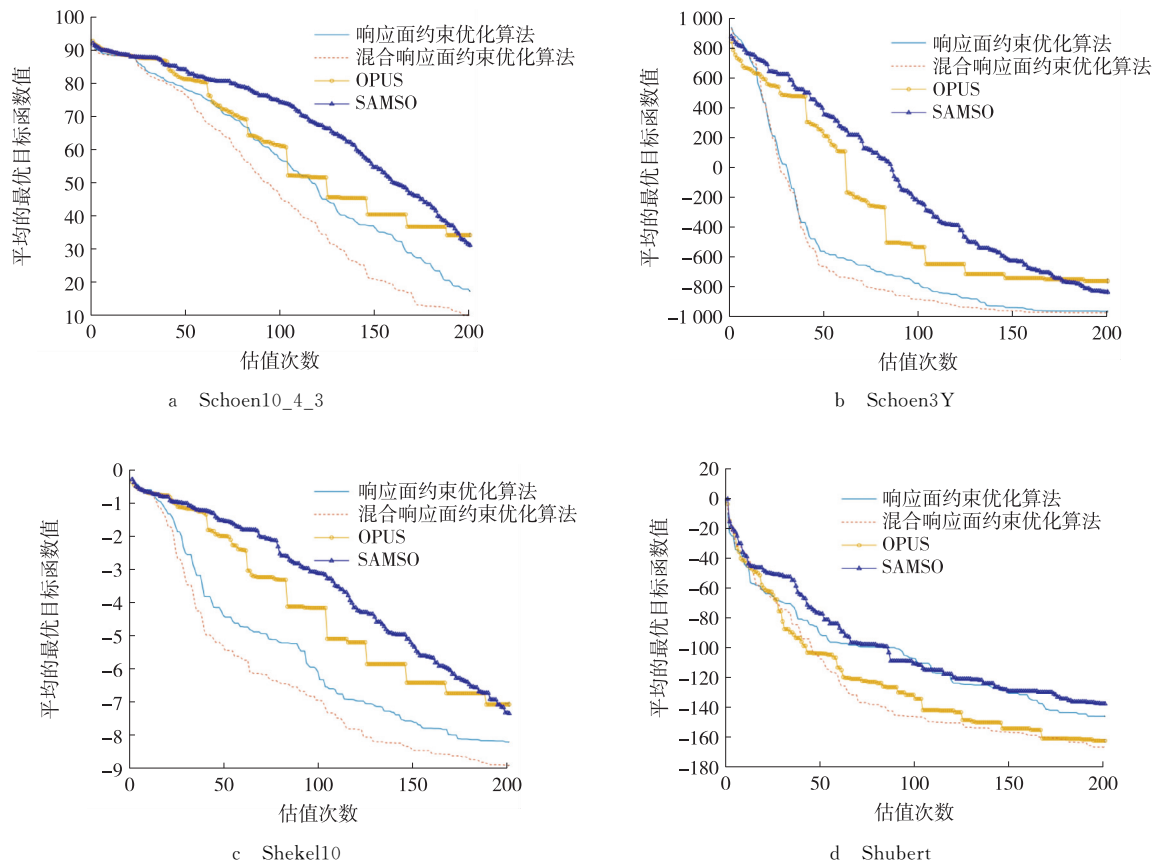


图 3 算法求得测试函数 30 次实验平均最优值变化曲线

Fig. 3 The curve of the variation of the average optimal value of 30 experiments is obtained by the algorithm

为了进一步比较 4 种算法的性能,记录了 4 种算法在 200 次函数估值后找到的最佳目标函数值的统计数据,即算法运行 30 次得到的 30 个函数最优值中的最小函数值、中位数、最大值、均值和标准误差,且上述数据均保留小数点后 2 位数值。由于篇幅限制,表 2 只呈现了其中的 7 个测试问题(Branin, Goldstein-Price, Hartman3, Hartman6, Shekel10, Shekel5 和 Shekel7)。从表 2 可知:如考虑最优值,则混合响应面约束优化算法在 7 个测试函数中的 3 个测试函数上的数值结果更好;如考虑中值,混合响应面约束优化算法在 5 个测试函数上的数值结果更好;如考虑均值,混合响应面约束优化算法在 5 个测试函数的结果更好;如考虑标准差,混合响应面约束优化算法在 4 个测试函数上的数值结果更好。综上所述,混合响应面约束优化算法不仅能够找到较为接近的黑箱函数真实最优值的近似值,并且模型的稳定性强于响应面约束优化算法、OPUS 算法和 SAMSO 算法(图 2 和表 2)。

#### 4 结束语

本文对基于响应面模型求解黑箱优化问题的一种径向基函数响应面算法(CORS)做了改进,从提高响应面模型的精度出发,选择两个径向基插值模型的组合作为算法的响应面模型,组合的初始权重因子人为设定,随着算法的迭代,权重因子进行自适应调整。数值实验结果说明了改进后算法的有效性。在今后的研究中,将继续探讨不同径向基插值模型的组合方式及对提高响应面算法效果的作用。

表 2 全局优化算法在 7 个测试函数上的比较

Tab. 2 Comparison of global optimization algorithms on seven test functions

| 测试函数            | 算法          | 最优值           | 中值            | 最大值          | 均值           | 标准差         |
|-----------------|-------------|---------------|---------------|--------------|--------------|-------------|
| Branin          | 响应面约束优化算法   | 0.40          | 0.40          | <b>0.40</b>  | <b>0.40</b>  | <b>0.00</b> |
|                 | 混合响应面约束优化算法 | 0.40          | 0.40          | <b>0.40</b>  | <b>0.40</b>  | <b>0.00</b> |
|                 | OPUS        | 0.40          | 0.40          | 0.57         | 0.41         | 0.03        |
|                 | SAMSO       | 0.40          | 0.40          | <b>0.40</b>  | <b>0.40</b>  | <b>0.00</b> |
| Goldstein_Price | 响应面约束优化算法   | 3.00          | <b>3.00</b>   | 5.27         | <b>3.08</b>  | 0.41        |
|                 | 混合响应面约束优化算法 | 3.00          | <b>3.00</b>   | 5.75         | 3.25         | 0.61        |
|                 | OPUS        | 3.00          | 3.08          | 18.90        | 4.60         | <b>3.97</b> |
|                 | SAMSO       | 3.00          | 3.06          | <b>5.18</b>  | 3.24         | 0.45        |
| Hartman3        | 响应面约束优化算法   | -3.86         | <b>-3.86</b>  | <b>-3.86</b> | <b>-3.86</b> | <b>0.00</b> |
|                 | 混合响应面约束优化算法 | -3.86         | <b>-3.86</b>  | -3.85        | <b>-3.86</b> | <b>0.00</b> |
|                 | OPUS        | -3.86         | -3.84         | -3.76        | -3.83        | 0.03        |
|                 | SAMSO       | -3.86         | <b>-3.86</b>  | -3.76        | -3.85        | 0.02        |
| Hartman6        | 响应面约束优化算法   | -3.32         | <b>-3.32</b>  | <b>-3.08</b> | -3.26        | 0.09        |
|                 | 混合响应面约束优化算法 | -3.32         | <b>-3.32</b>  | -3.05        | <b>-3.28</b> | 0.09        |
|                 | OPUS        | -3.32         | -3.24         | -3.02        | -3.22        | 0.09        |
|                 | SAMSO       | -3.32         | -3.31         | -3.06        | -3.26        | <b>0.08</b> |
| Shekel10        | 响应面约束优化算法   | <b>-10.54</b> | -10.42        | <b>-2.81</b> | -8.17        | 3.10        |
|                 | 混合响应面约束优化算法 | <b>-10.54</b> | <b>-10.45</b> | <b>-2.81</b> | <b>-8.88</b> | <b>2.49</b> |
|                 | OPUS        | -10.50        | -7.87         | <b>-2.81</b> | -7.25        | 2.63        |
|                 | SAMSO       | -10.42        | -8.08         | -2.34        | -7.42        | 2.56        |
| Shekel5         | 响应面约束优化算法   | -10.14        | -6.77         | <b>-2.68</b> | -7.18        | 2.55        |
|                 | 混合响应面约束优化算法 | <b>-10.15</b> | <b>-8.06</b>  | <b>-2.68</b> | <b>-7.41</b> | <b>2.40</b> |
|                 | OPUS        | -9.87         | -2.66         | -2.30        | -4.49        | 2.88        |
|                 | SAMSO       | -9.95         | -4.65         | -2.43        | -5.27        | 2.77        |
| Shekel7         | 响应面约束优化算法   | <b>-10.40</b> | <b>-10.37</b> | -3.09        | <b>-8.96</b> | <b>2.48</b> |
|                 | 混合响应面约束优化算法 | <b>-10.40</b> | -9.86         | <b>-3.72</b> | -8.37        | 2.70        |
|                 | OPUS        | -10.29        | -7.40         | -2.51        | -6.86        | 2.68        |
|                 | SAMSO       | -10.23        | -7.71         | -2.67        | -6.63        | 2.59        |

注:加粗数值表示在同类数据中占优

## 参考文献:

- [1] BJÖRKMAN M, HOLMSTRÖM K. Global optimization of costly nonconvex functions using radial basis functions[J]. Optimization and Engineering, 2000, 1(4): 373-397.
- [2] MYERS R H, MONTGOMERY D C. Response surface methodology: process and product optimization using designed experiments[M]. New York: John Wiley & Sons, Inc., 1995.
- [3] BHOSSEKAR A, IERAPETRITOU M. Advances in surrogate based modeling, feasibility analysis, and optimization: a review[J]. Computers & Chemical Engineering, 2018, 108: 250-267.
- [4] GUTMANN H M. A radial basis function method for global optimization[J]. Journal of Global Optimization, 2001, 19(3): 201-227.
- [5] 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.

- [6] REGIS R G, SHOEMAKER C A. A stochastic radial basis function method for the global optimization of expensive functions [J]. *Inform Journal on Computing*, 2007, 19(4): 497-509.
- [7] ACAR E. Various approaches for constructing an ensemble of metamodels using local measures [J]. *Structural and Multidisciplinary Optimization*, 2010, 42(6): 879-896.
- [8] REGIS R G. Particle swarm with radial basis function surrogates for expensive black-box optimization [J]. *Journal of Computational Science*, 2014, 5(1): 12-23.
- [9] JIE H X, WU Y Z, DING J W. An adaptive metamodel-based global optimization algorithm for black-box type problems [J]. *Engineering Optimization*, 2015, 47(11): 1459-1480.
- [10] MÜLLER J. SOCEMO: surrogate optimization of computationally expensive multiobjective problems [J]. *Inform Journal on Computing*, 2017, 29(4): 581-596.
- [11] 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.
- [12] MÜLLER J, DAY M. Surrogate optimization of computationally expensive black-box problems with hidden constraints [J]. *Inform Journal on Computing*, 2019, 31(4): 689-702.
- [13] LI F, CAI X W, GAO L, et al. A surrogate-assisted multiswarm optimization algorithm for high-dimensional computationally expensive problems [EB/OL]. (2020-02-11)[2020-11-17]. <https://ieeexplore.ieee.org/document/8994184/authors#authors>.
- [14] YE K Q, LI W, SUDJANTO A. Algorithmic construction of optimal symmetric Latin hypercube designs [J]. *Journal of Statistical Planning and Inference*, 2000, 90(1): 145-159.
- [15] DIXON L C W, SZEGÖ G P. The global optimization problem: an introduction [M]//DIXON L C W, SZEGÖ G P. *Towards global optimization 2*. Amsterdam: North-Holland Publishing Co., 1978: 1-15.
- [16] TÖRN A, ZILINSKAS A. *Global optimization* [M]. Heidelberg: Springer-Verlag, 1989.
- [17] SCHOEN F. A wide class of test functions for global optimization [J]. *Journal of Global Optimization*, 1993, 3(2): 133-137.
- [18] SURJANOVIC S, BINGHAM D. Virtual library of simulation experiments: test functions and datasets [EB/OL]. [2020-11-17]. <http://www.sfu.ca/~ssurjano/optimization.html>.
- [19] DOLAN E D, MORÉ J J. Benchmarking optimization software with performance profiles [J]. *Mathematical Programming*, 2002, 91(2): 201-213.
- [20] 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

### A Hybrid Radial Basis Function Response Surface Method Based on Constrained Optimization

YIN Yue, BAI Fusheng

(School of Mathematical Sciences, Chongqing Normal University, Chongqing 401331, China)

**Abstract:** [Purposes] To better solve the expensive black box function optimization problem. [Methods] Based on the constrained optimization by using response surfaces method, a hybrid model obtained by combining the thin plate spline radial basis function model and the cubic radial basis function model is used as a new response surface model to approximate the black-box function, and during the optimizing process the weight factors of the hybrid model are adaptively selected. [Findings] The hybrid radial basis function response surface algorithm is constructed. [Conclusions] Numerical results of the algorithm on 34 common test problems show that the proposed algorithm is better than the original constrained optimization by using response surfaces algorithm, the OPUS algorithm and the SAMSO algorithm.

**Keywords:** expensive black-box function; global optimization; response surface method; radial basis function

(责任编辑 方 兴)