Journal of Global Optimization (2020) 77:919–948
https://doi.org/10.1007/s10898-020-00916-w

Finding optimal points for expensive functions using adaptive RBF-based surrogate model via uncertainty quantification

Ray-Bing Chen[1] · **Yuan Wang**[2] · **C. F. Jeff Wu**[3]

Received: 1 December 2018 / Accepted: 24 May 2020 / Published online: 9 June 2020

Abstract
Global optimization of expensive functions has important applications in physical and computer experiments. It is a challenging problem to develop efficient optimization scheme, because each function evaluation can be costly and the derivative information of the function is often not available. We propose a novel global optimization framework using adaptive radial basis functions (RBF) based surrogate model via uncertainty quantification. The framework consists of two iteration steps. It first employs an RBF-based Bayesian surrogate model to approximate the true function, where the parameters of the RBFs can be adaptively estimated and updated each time a new point is explored. Then it utilizes a model-guided selection criterion to identify a new point from a candidate set for function evaluation. The selection criterion adopted here is a sample version of the expected improvement criterion. We conduct simulation studies with standard test functions, which show that the proposed method has some advantages, especially when the true function has many local optima. In addition, we also propose modified approaches to improve the search performance for identifying optimal points.

Keywords Expected improvement · Markov chain Monte Carlo · Radial basis functions · Sequential design

1 Introduction

In this paper, we consider the problem of global optimization of expensive functions, i.e., functions which require large computational costs to evaluate. For physical and computational

Ray-Bing Chen, Yuan Wang: Joint first authors.

✉ C. F. Jeff Wu
jeff.wu@isye.gatech.edu

[1] Department of Statistics, National Cheng Kung University, Tainan, Taiwan

[2] Wells Fargo, Minneapolis, USA

[3] H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, USA

experiments, these functions represent the relationship between input and output variables, and may require days or even weeks to evaluate at a single input setting. One example is the high-pressure mixing and combustion processes in liquid rocket engines, which requires numerically solving a large, coupled system of partial differential equations; see Oefelein and Yang [18]. Even when computation is parallelized over thousands of processing cores, a comprehensive simulation of a single injector may take months to complete. An important problem about expensive functions is how to optimize the output/response by choosing appropriate settings of the input variables. This problem can be challenging for two reasons. First, it is not feasible to conduct extensive runs of function evaluations to find the optimal input settings, since each function evaluation is expensive. It is thus desirable to identify the optimal input settings with as few runs as possible. The second challenge comes from the complicated nature of the functional relationship. They are usually regarded as "black-boxes", because there is no explicit relationship between the input and output. Although various local optimization methods are available when the derivatives of the functions are known or can be easily obtained, see Boyd and Vandenberghe [3], such methods are not applicable in the present scenario.

In the literature, a widely used practice for global optimization of expensive functions is to sequentially select input settings for function evaluations based on some criterions. The approach consists of two steps. First, it constructs a surrogate model to approximate the true function based on all the observed function outputs. The advantage of employing surrogate model is that it can provide predictions at any input settings with much cheaper computation. Second, it identifies a new input setting for function evaluation according to some surrogate model based selection criteria. With this approach, it is feasible to approximate the global optimizer of the true function via the surrogate model optimization. The commonly used surrogate models are the kriging models [12,13] and models based on radial basis functions [11,20]. Chen et al. [5,6] proposed to construct the surrogate models via overcomlpete pre-specified basis functions. In addition, another type of optimization approach is statistical global optimization which chooses the next point based on a probability improvement criterion, like P-algorithm [23]. Gutmann [11] and Žilinskas [24] have showed the equivalence of the P-algorithm and the surrogate approach proposed in Gutmann [11] under certain conditions. For more details along these lines, see a review in Žilinskas [25].

The primary objective of this paper is to propose a novel global optimization framework for optimizing expensive functions. Our approach is motivated by Regis and Shoemaker [20], in which they utilize Radial Basis Functions (RBF) to build a deterministic surrogate model and guide the selection of the next explored point based on the predicted response and some distance criteria. The rationale of using RBFs is that they can capture the nonlinear trend of functions. However, the RBFs they used are pre-determined and lack the flexibility of modeling. Also, it is less efficient to perform function evaluation from their surrogate model, because they use RBFs in an interpolation way without providing prediction uncertainties. Although a distance criterion is used to avoid getting trapped at local optima, it does not incorporate the information in prediction uncertainty for the surrogate models. To make better use of all information in the data, we propose to construct surrogate model with RBFs that are chosen adaptively based on the updated outputs, and to select new points based on surrogate models with quantified uncertainties.

There are other approaches for global optimization of expensive functions in the literature. Jones et al. [13] propose a global optimization scheme by constructing a surrogate model with the kriging method. Our approach is different in that they make strong assumptions on the correlation structure between explored points while ours does not. A detailed review related to the kriging model in global optimization can be found in Jones [12]. Chen et al. [6]

propose a global optimization scheme that builds a mean prediction model with linear basis functions selected from a dictionary of functions, and then imposes a Bayesian structure over the mean model to quantify the uncertainty of the prediction. Our approach is also different from Chen et al. [6]. Instead of using a predetermined discrete function dictionary with a large number of linear functions, we use a moderate number of RBFs that can be adaptively updated based on observed data.

The paper is organized as follows. In Sect. 2, we give a mathematical formulation of the global optimization problem, and provide a review of the RBFs. In Sect. 3, we present the proposed global optimization framework that utilizes adaptive RBF-based Bayesian surrogate model. In Sect. 4, we present simulation studies to validate and compare our proposed method with the methods by Regis and Shoemaker [20] and Jones et al. [13]. A modification of the proposed method to avoid getting trapped in local optima is presented in Sect. 5.1. In addition, we study the effect of the grid size which is used as the candidate points in the proposed method. Concluding remarks and future research directions are given in Sect. 6.

2 Problem formulation and review of RBFs

Suppose $f(\mathbf{x})$ is an expensive function of interest, where $\mathbf{x} = (x^1, \ldots, x^p)^T \in V$, and V is a p-dimensional convex domain in R^p. The objective is to identify an optimal input setting $\mathbf{x}_{opt}$ that maximizes $f(\mathbf{x})$,

$$\mathbf{x}_{opt} = \arg\max_{\mathbf{x}\in V} f(\mathbf{x}). \tag{1}$$

Because it is not practical to evaluate $f(\mathbf{x})$ over V to search the global maximizer due to the huge computational cost, a well-established practice is to sequentially select a few input settings for function evaluation using a two-step strategy. Suppose a set of N function evaluations $\{(\mathbf{x}_i, f(\mathbf{x}_i))\}_{i=1}^N$ are taken. In step 1, a surrogate model is constructed and the resulting model approximation is denoted by $f_N(\mathbf{x})$. Unlike the true function $f(\mathbf{x})$, the surrogate model is much cheaper to build and evaluate. Therefore it is feasible to predict function values over all $\mathbf{x} \in V$.

In step 2, the next input setting $\mathbf{x}_{N+1}$ is selected for function evaluation via certain criterion based on the surrogate model from step 1. Steps 1 and 2 iterate until the total computational budget is met. The best point among all the chosen input settings, $\hat{\mathbf{x}}_{opt} = \arg\max_{\mathbf{x}_i} f(\mathbf{x}_i)$, can be treated as an approximation to the true optimal point $\mathbf{x}_{opt}$.

By adopting this two-step strategy, we will present in Sect. 3 our proposed framework in detail. Note that the surrogate construction may not necessarily be an interpolator of the observed points, i.e., $f_N(\mathbf{x}_i) \neq f(\mathbf{x}_i)$. Because our goal is optimization, the surrogate is used to predict the location of the optimal points, rather than to approximate the response with high accuracy [5]. Thus we want to capture the trend of the true response surface quickly and to serve this purpose, our surrogate model does not have to meet the interpolation requirement.

In the remaining part of this section, we give a brief review of the RBFs, which will be used in the proposed framework for the surrogate model construction. In the literature, the RBF is popularly deployed in applied mathematics and neural networks. See Buhmann [4] and Bishop [2]. Several commonly used functions are: (1) Gaussian functions: $r(\mathbf{x}; \boldsymbol{\mu}, s) = \exp\{-s^2||\mathbf{x}-\boldsymbol{\mu}||^2\}$; (2) generalized multi-quadratic functions: $r(\mathbf{x}; \boldsymbol{\mu}, t) = (||\mathbf{x}-\boldsymbol{\mu}||^2+t^2)^\delta$ with $t > 0, 0 < \delta < 1$; (3) generalized inverse multi-quadratic functions: $r(\mathbf{x}; \boldsymbol{\mu}, t) =$

$(||\mathbf{x}-\boldsymbol{\mu}||^2+t^2)^{-\delta}$ with $t>0, \delta>0$; (4) thin plate spline functions: $r(\mathbf{x};\boldsymbol{\mu})=||\mathbf{x}-\boldsymbol{\mu}||^2\ln(||\mathbf{x}-\boldsymbol{\mu}||)$, where $\boldsymbol{\mu}$ is the center of the function, and s and t are pre-specified constants which vary with the chosen function.

In our work, we will focus on the Gaussian RBFs. The Gaussian RBFs have two types of parameters: the center parameter $\boldsymbol{\mu}\in V$ that determines the location of the RBFs, and the scale parameter s that measures the degree of fluctuation of the function. One advantage of using the Gaussian RBFs over other basis functions is that it can capture different trends of response by choosing different centers and scales. For example, a larger s indicates a more concentrated change in the surface, and vice versa.

3 General global optimization framework

In this section, we propose a global optimization framework that utilizes adaptive RBF-based surrogate model via uncertainty quantification. In Sect. 3.1, we propose a novel hierarchical normal mixture Bayesian surrogate model with RBFs to approximate the true function, where the model coefficients are sparsely represented to avoid over-fitting, and the parameters of the RBFs are adaptively updated each time a new point is explored. This allows us to predict the function value at any given candidate point. In Sect. 3.2, we propose a model-guided selection criterion and based on the posterior samples, a sample version of the expected improvement criterion is adopted. A new point can then be selected to identify a more promising area of global maximizer. A summary of the algorithm and some discussions will be presented in Sect. 3.3.

3.1 Normal mixture surrogate model with RBFs

Suppose we observe N explored points $\mathcal{P}_{exp}=\{\mathbf{x}_1,\ldots,\mathbf{x}_N\}$, and its function values $\mathbf{y}=(y_1,\ldots,y_N)^T=(f(\mathbf{x}_1),\ldots,f(\mathbf{x}_N))^T$. Without loss of generality, we assume $E(y_i)=0$, because otherwise we can approximate $(y_i-\bar{y})$'s instead of y_i's, where $\bar{y}$ is the sample mean of $y_1,\ldots,y_N$, i.e., $\bar{y}=\sum_{i=1}^N y_i/N$. We propose to construct a surrogate model by a summation of N Gaussian RBFs $r(\mathbf{x};\boldsymbol{\mu}_i,s_i)=\exp\{-s_i^2||\mathbf{x}-\boldsymbol{\mu}_i||^2\}$ and an error term $\epsilon(\mathbf{x})$:

$$f(\mathbf{x})=f_N(\mathbf{x})+\epsilon(\mathbf{x})=\sum_{i=1}^{N}\beta_i r(\mathbf{x};\boldsymbol{\mu}_i,s_i)+\epsilon(\mathbf{x}). \tag{2}$$

Here, an error term is used to model the discrepancy between the model approximation $f_N(\mathbf{x})$ constructed by the RBFs and the true function $f(\mathbf{x})$. We assume that $\epsilon(\mathbf{x})$ are independent normal distributions with mean 0 and vairance σ^2. Note that if the center parameters $\boldsymbol{\mu}_i$'s and the scale parameters s_i's are known and fixed, then the surrogate model in (2) is exactly the same as linear regression.

3.1.1 Prior distributions

Because both $\boldsymbol{\mu}_i$'s and s_i's are unknown, the proposed modeling approach can handle highly nonlinear functions. A uniform prior over a rectangular region is used for $\boldsymbol{\mu}=(\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_N)$,

$$\boldsymbol{\mu}_i\sim\text{Uniform}(\Omega),\quad i=1,\ldots,N, \tag{3}$$

where $\Omega = \prod_{j=1}^{p}[\min(x_{1:N}^{j}), \max(x_{1:N}^{j}))]$, whose hypervolume is $Vol(\Omega)$. Ω denotes the smallest hyper-rectangle to cover the current explored points, and it is adaptively changed with the addition of new explored points, see Andrieu et al. [1]. A gamma prior is used for the scale parameters $\mathbf{s} = (s_1, \ldots, s_N)^T$,

$$s_i \sim \text{Gamma}(a_s, b_s), \tag{4}$$

where a_s and b_s are common to all i's.

We also impose a hierarchical structure on the coefficients β_i's. Define a latent variable $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_N)^T$ to indicate whether a certain basis function is important or not: $\gamma_i = 1$ indicates that the ith basis is important, while $\gamma_i = 0$ indicates the opposite. Specifically, we set $\beta_i|(\gamma_i = 0) \sim N(0, \tau_i)$ with small τ_i, and $\beta_i|(\gamma_i = 1) \sim N(0, C\tau_i)$ with relatively large C, where C can be interpreted as a variance ratio. This hierarchical setting is first employed in the Stochastic Search Variable Selection (SSVS) scheme by George and McCulloch [10] and is also used for uncertainty quantification studies in Chen et al. [6]. Indeed, it is one type of the "g-prior" (see Zellner [26]) for avoiding over-fitting. Now the mixture normal prior of the model coefficient $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_N)^T$ can be written as follows:

$$\boldsymbol{\beta}|\boldsymbol{\gamma} \sim N(0, \Sigma_\tau^2), \text{ where } \Sigma_\tau = \text{diag}(a_1\tau_1, \ldots, a_N\tau_N), \tag{5}$$

with $a_i = 1$ if $\gamma_i = 0$ and $= C$ if $\gamma_i = 1$, and a binomial prior for the latent variable γ_i,

$$P(\gamma_i = 0) = p_i, P(\gamma_i = 1) = 1 - p_i, \quad i = 1, \ldots, N. \tag{6}$$

Note that the choice of C plays an important role in the posterior sampling and control the model complexity. We also impose an inverse-gamma prior for the residual variance σ^2,

$$\sigma^2 \sim IG\left(\frac{\nu_0}{2}, \frac{\zeta_0}{2}\right). \tag{7}$$

By combining (2)–(7) with independent prior assumptions, we obtain the full posterior distribution of $\{\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\gamma}, \sigma^2, \mathbf{s}\}$

$$\begin{aligned} p(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\gamma}, \sigma^2, \mathbf{s}|\mathcal{P}_{exp}, \mathbf{y}) &\propto p(\mathbf{y}|\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\gamma}, \sigma^2, \mathbf{s}, \mathcal{P}_{exp}) \cdot p(\boldsymbol{\beta}|\boldsymbol{\gamma}, \boldsymbol{\mu}) \cdot p(\boldsymbol{\gamma}) \cdot p(\mathbf{s}) \cdot p(\boldsymbol{\mu}) \cdot p(\sigma^2) \\ &= \left[(2\pi\sigma^2)^{-N/2} \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{y} - D(\boldsymbol{\mu}, \mathbf{s}) \cdot \boldsymbol{\beta})^T(\mathbf{y} - D(\boldsymbol{\mu}, \mathbf{s}) \cdot \boldsymbol{\beta})\right\}\right] \left[\prod_{i=1}^{N+p} p_i^{\gamma_i}(1 - p_i)^{(1-\gamma_i)}\right] \\ &\quad \left[\det(2\pi\,\Sigma_\tau^2)^{-1/2} \exp\left\{-\frac{1}{2}\boldsymbol{\beta}^T\Sigma_\tau^{-2}\boldsymbol{\beta}\right\}\right] \prod_{i=1}^{N}\left[\frac{b_s^{a_s}}{\Gamma(a_s)} s_i^{a_s-1} \exp(-b_s s_i)\right] \left[\frac{1_\Omega(\boldsymbol{\mu}_{1:N})}{Vol(\Omega)}\right] \\ &\quad \left[(\sigma^2)^{-(\nu_0/2+1)} \exp\left\{-\frac{\zeta_0}{2\sigma^2}\right\}\right], \end{aligned} \tag{8}$$

where the coefficient matrix $D(\boldsymbol{\mu}, \mathbf{s})$ is defined as

$$D(\boldsymbol{\mu}, \mathbf{s}) = \begin{pmatrix} r(\mathbf{x}_1; \boldsymbol{\mu}_1, s_1) & \cdots & r(\mathbf{x}_1; \boldsymbol{\mu}_N, s_N) \\ \vdots & \ddots & \vdots \\ r(\mathbf{x}_N; \boldsymbol{\mu}_1, s_1) & \cdots & r(\mathbf{x}_N; \boldsymbol{\mu}_N, s_N) \end{pmatrix},$$

and the indicator function $1_\Omega(\boldsymbol{\mu}) = 1$ if $\boldsymbol{\mu} \in \Omega$, $= 0$ if $\boldsymbol{\mu} \notin \Omega$.

3.1.2 Posterior sampling

The posterior distribution defined in (8) is computationally intractable. Markov Chain Monte Carlo (MCMC) method is utilized to solve this problem, see Andrieu et al. [1] and Koutsourelakis [14]. That is, we use the MCMC method to generate the posterior samples from $p(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\gamma}, \sigma^2, \mathbf{s}|\mathcal{P}_{exp}, \mathbf{y})$. Thus we sequentially sample $\boldsymbol{\beta}, \boldsymbol{\gamma}, \sigma^2, \boldsymbol{\mu}$ and $\mathbf{s}$ by fixing the other components and the data $\{\mathcal{P}_{exp}, \mathbf{y}\}$. Under certain conditions, we can guarantee that these samples can be treated as the posterior samples of $\boldsymbol{\beta}, \boldsymbol{\gamma}, \sigma^2, \boldsymbol{\mu}, \mathbf{s}$. Here the MCMC method iterates the following two steps:

- Sample $\boldsymbol{\beta}, \boldsymbol{\gamma}, \sigma^2$ by fixing $\boldsymbol{\mu}, \mathbf{s}, \mathcal{P}_{exp}$ and $\mathbf{y}$.
- Sample $\boldsymbol{\mu}$ and $\mathbf{s}$ by fixing $\boldsymbol{\beta}, \boldsymbol{\gamma}, \sigma^2, \mathcal{P}_{exp}$, and $\mathbf{y}$.

Specifically, we use the Gibbs sampler to generate the posterior samples for the parameters $\boldsymbol{\beta}, \boldsymbol{\gamma}, \sigma^2$, and the Metropolis–Hasting algorithm to obtain the posterior samples for the parameters $\boldsymbol{\mu}$ and $\mathbf{s}$, because there is no explicit formula for the posterior distributions of $\boldsymbol{\mu}$ and $\mathbf{s}$.

Start with the posterior distributions for $\boldsymbol{\beta}, \boldsymbol{\gamma}, \sigma^2$. Denote $M = (D(\boldsymbol{\mu}, \mathbf{s})^T D(\boldsymbol{\mu}, \mathbf{s})/\sigma^2 + \Sigma_\tau^{-2})^{-1}$, and $h = MD(\boldsymbol{\mu}, \mathbf{s})^T \mathbf{y}/\sigma^2$. Then, the samples of $\boldsymbol{\gamma}$ can be generated by

$$\boldsymbol{\beta}|\boldsymbol{\mu}, \sigma^2, \boldsymbol{\gamma}, \mathbf{s}, \mathcal{P}_{exp}, \mathbf{y} \sim \mathrm{N}(h, M). \tag{9}$$

The samples of σ^2 can be generated by

$$\sigma^2|\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\gamma}, \mathbf{s}, \mathcal{P}_{exp}, \mathbf{y} \sim \mathrm{IG}\left(\frac{\nu_0 + N}{2}, \frac{\zeta_0 + |\mathbf{y} - D(\boldsymbol{\mu}, \mathbf{s})\boldsymbol{\beta}|^2}{2}\right). \tag{10}$$

For the samples of $\boldsymbol{\gamma}$, it would be simple to sample γ_i sequentially conditional on the other components, and γ_i can be generated by

$$P(\gamma_i = 1|\boldsymbol{\beta}, \boldsymbol{\mu}, \mathbf{s}, \sigma, \gamma_{-i}, \mathcal{P}_{exp}, \mathbf{y}) = p_1/(p_1 + p_0), \tag{11}$$

where

$$p_1 = p(\boldsymbol{\beta}|\gamma_i = 1, \gamma_{-i}, \boldsymbol{\mu}, \mathbf{s})p(\gamma_i = 1, \gamma_{-i}) \propto \det(\Sigma^*)^{-1/2} \exp\left\{-\frac{1}{2}\boldsymbol{\beta}^T(\Sigma^*)^{-1}\boldsymbol{\beta}\right\}(1 - p_i)$$

with $\Sigma^* = D_r^{i+}$, and D_r^{i+} is Σ_τ with $\gamma_i = 1$;

$$p_0 = p(\boldsymbol{\beta}|\gamma_i = 0, \gamma_{-i}, \boldsymbol{\mu}, \mathbf{s})p(\gamma_i = 0, \gamma_{-i}) \propto \det(\Sigma^*)^{-1/2} \exp\left\{-\frac{1}{2}\boldsymbol{\beta}^T(\Sigma^*)^{-1}\boldsymbol{\beta}\right\} p_i$$

with $\Sigma^* = D_r^{i-}$, and D_r^{i-} is Σ_τ with $\gamma_i = 0$. Here the notation $\gamma_{-i} = (\gamma_1, \ldots, \gamma_{i-1}, \gamma_{i+1}, \ldots, \gamma_N)^T$ represents the vector of all γ_j's except γ_i.

Now we turn to the parameters $\boldsymbol{\mu}$ and s. First, consider the sampling procedure for $\boldsymbol{\mu}$. Instead of directly sampling the vector $\boldsymbol{\mu}$, we suggest sampling $\boldsymbol{\mu}_i$ sequentially from

$$\begin{aligned} &p(\boldsymbol{\mu}_i|\boldsymbol{\mu}_{-i}, \boldsymbol{\beta}, \mathbf{s}, \sigma, \mathcal{P}_{exp}, \mathbf{y}) \\ &\quad\propto \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{y} - D(\boldsymbol{\mu}, \mathbf{s})\boldsymbol{\beta})^T(\mathbf{y} - D(\boldsymbol{\mu}, \mathbf{s})\boldsymbol{\beta})\right\} 1_\Omega(\boldsymbol{\mu}_{1:N}), \end{aligned} \tag{12}$$

where $\boldsymbol{\mu}_{-i} = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{i-1}, \boldsymbol{\mu}_{i+1}, \ldots, \boldsymbol{\mu}_N)$ denotes the vector of all $\boldsymbol{\mu}_j$'s except $\boldsymbol{\mu}_i$. We use the Metropolis–Hasting algorithm to generate posterior samples for $\boldsymbol{\mu}_i$. Specifically, at a new step $(k+1)$, we set the proposed density to be a mixture of two densities, and a temporary

sample $\boldsymbol{\mu}_i^*$ can be obtained from the whole domain Ω with uniform probability, or it can be a perturbation of the current iteration $\boldsymbol{\mu}_i^{(k)}$ within its local neighborhood, i.e.,

$$q_1(\boldsymbol{\mu}_i^*) = \text{Uniform}(\Omega), \text{ with probability } \omega,$$
$$\text{and } q_2(\boldsymbol{\mu}_i^*) = N(\boldsymbol{\mu}_i^{(k)}, \sigma_\mu^2) \text{ with probability } 1-\omega. \tag{13}$$

Then we accept this temporary sample $\boldsymbol{\mu}_i^*$ with the acceptance rate

$$A(\boldsymbol{\mu}_i, \boldsymbol{\mu}_i^*) = \min\{1, \left(\frac{\exp\{-1/(2\sigma^2)|\mathbf{y} - D(\boldsymbol{\mu}^*, \mathbf{s})\boldsymbol{\beta}|^2\}}{\exp\{-1/(2\sigma^2)|\mathbf{y} - D(\boldsymbol{\mu}, \mathbf{s})\boldsymbol{\beta}|^2\}}\right) 1_\Omega(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_i^*, \ldots, \boldsymbol{\mu}_N)\}$$

where $\boldsymbol{\mu}^* = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_i^*, \ldots, \boldsymbol{\mu}_N)^T$.

Similarly, we can use the Metropolis–Hasting algorithm to generate samples of s_i. At step $(k+1)$, we choose a temporary s_i^* as a perturbation of the current sample $s_i^{(k)}$ by the proposed density

$$q_3(s_i^*) = N(s_i^{(k)}, \sigma_s^2). \tag{14}$$

And we accept such sample s_i^* with the acceptance rate

$$A(s_i, s_i^*) = \min\left\{1, \left(\frac{\exp\{-1/(2\sigma^2)|\mathbf{y} - D(\boldsymbol{\mu}, \mathbf{s}^*)\boldsymbol{\beta}|^2\}}{\exp\left\{-1/(2\sigma^2)|\mathbf{y} - D(\boldsymbol{\mu}, \mathbf{s})\boldsymbol{\beta}|^2\right\}} \cdot \frac{(s_i^*)^{a_s-1}\exp(-b_s s_i^*)}{s_i^{a_s-1}\exp(-b_s s_i)}\right)\right\}$$

where $\mathbf{s}^* = (s_1, \ldots, s_i^*, \ldots, s_N)$.

From (9)–(14), we generate samples for $\boldsymbol{\gamma}$, $\boldsymbol{\beta}$, σ, $\boldsymbol{\mu}$, $\mathbf{s}$ iteratively based the updated estimate for the remaining parameters. Then, the Gibbs sequence,

$$\boldsymbol{\gamma}^{(0)}, \boldsymbol{\beta}^{(0)}, \sigma^{(0)}, \boldsymbol{\mu}^{(0)}, \mathbf{s}^{(0)}, \ldots, \boldsymbol{\gamma}^{(k)}, \boldsymbol{\beta}^{(k)}, \sigma^{(k)}, \boldsymbol{\mu}^{(k)}, \mathbf{s}^{(k)}, \ldots, \boldsymbol{\gamma}^{(K)}, \boldsymbol{\beta}^{(K)}, \sigma^{(K)}, \boldsymbol{\mu}^{(K)}, \mathbf{s}^{(K)},$$

can be obtained, where K is the total number of iterations. After discarding the first say 40% samples, the remaining samples can be treated as the posterior samples of $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$, σ^2, $\boldsymbol{\mu}$ and $\mathbf{s}$ from $p(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\gamma}, \sigma^2, \mathbf{s}|\mathcal{P}_{exp}, \mathbf{y})$. Thus the posterior sample $f_N^{(k)}(\tilde{\mathbf{x}})$ for model approximation at a candidate explored point $\tilde{\mathbf{x}}$ can be calculated by

$$f_N^{(k)}(\tilde{\mathbf{x}}) = \sum_{i=1}^{N} \beta_i^{(k)} r(\tilde{\mathbf{x}}; \boldsymbol{\mu}_i^{(k)}, s_i^{(k)}). \tag{15}$$

Then the function prediction $f_N(\tilde{\mathbf{x}})$ can then be calculated as the average of $f_N^{(k)}(\tilde{\mathbf{x}})$'s, and the prediction uncertainties can be calculated as the sample variance of $f_N^{(k)}(\tilde{\mathbf{x}})$'s.

Finally, we note that the mean value of the posterior density of $\boldsymbol{\beta}$ in (9) is $h = ((D(\boldsymbol{\mu}, \mathbf{s})^T \cdot D(\boldsymbol{\mu}, \mathbf{s})/\sigma^2 + \Sigma_\tau^{-2})^{-1})D(\boldsymbol{\mu}, \mathbf{s})^T\mathbf{y}/\sigma^2$, which is a biased estimator of $\boldsymbol{\beta}$ with a nugget value Σ_τ^{-2}. Hence, this estimate of $\boldsymbol{\beta}$ can be regarded as a ridge-type regression estimate. It is deployed to prevent the model coefficients from being too large. Its use can lead to a more stable surrogate model.

3.1.3 Tuning parameters

A remaining issue in the Bayesian computation is the tuning of the hyper-parameters, which is critical for the model performance. For the hyper-parameters related to the RBF, we adopt the settings in Andrieu et al. [1] and Koutsourelakis [14]. Specifically, for the proposed density of the RBF centers $\boldsymbol{\mu}_i$ in (13), we set $\sigma_\mu^2 = 0.001$. For the prior of the RBF scales s_i

in (4), we set $a_s = 2, b_s = 0$, and for the proposed density of s_i in (14), we set $\sigma_s^2 = 0.5$. For the hyper-parameters related to model coefficients and residuals, we follow the settings in Chipman et al. [7]. Specifically, for τ_i, we suggest to set $\tau_i = \Delta\mathbf{y}/(3\Delta\mathbf{x})$, where $\Delta\mathbf{x} = \max(\mathbf{x}_{1:N}^{1:p}) - \min(\mathbf{x}_{1:N}^{1:p})$, i.e., the largest change in $\mathbf{x}_{1:N}$, and $\Delta\mathbf{y} = \sqrt{Var(\mathbf{y})}/5$. For the prior of the indicator variable γ_i, we set $p_i = 0.5$, i.e., the probability of selecting a variable is 50%. For the hyper parameter ν_0 and γ_0 in (7), we set $\nu_0 = 2$, and $\nu_0\gamma_0$ to be the 99% quantile of the inverse gamma prior that is close to $\sqrt{Var(\mathbf{y})}$. Consider the variance ratio C. Usually we choose a large positive value for C, e.g., $C \geq 10$. From our experience, we fix $C = 25$ in the first simulation example.

3.2 A point selection criterion

In this section, we discuss how to select new explored points based on the uncertainty of the response prediction for exploring uncertain regions. The ideal selection criterion should perfectly balance between exploration and exploitation properties to efficiently identify the optimal points within the given search budget. Here a sample version of the Expected Improvement criterion is adopted.

The EI criterion, initially proposed by Mockus et al. [16], is used to select points close to the global maxima based on a chosen surrogate model. Using this criterion, an explored point is selected to maximize the expected improvement over the best observed response

$$E(I(\mathbf{x})) = E(\max\{y - f_{\max}, 0\}), \tag{16}$$

where $f_{\max} = \max\{y_1, \ldots, y_N\}$ is the maximum of the observed model outputs. It is pointed out in Jones et al. [13] that under the Gaussian assumption of $y \sim N(\mu_0, s_0^2)$, $E(I(\mathbf{x}))$ has the following closed form expression:

$$E(I(\mathbf{x})) = (\mu_0 - f_{\max})\Phi\left(\frac{\mu_0 - f_{\max}}{s_0}\right) + s_0\phi\left(\frac{\mu_0 - f_{\max}}{s_0}\right). \tag{17}$$

By examining the terms, we see that the expected improvement is large for those $\mathbf{x}$ having either (i) a predicted value at $\mathbf{x}$ that is much larger than the maximum of outputs obtained so far, i.e., $\mu_0 \gg f_{\max}$, or (ii) having much uncertainty about the value of $y(\mathbf{x})$, i.e., when s_0 is large.

In our scenario, since the proposed surrogate model does not satisfy the Gaussian assumption, there is no analytical form for y, and thus it is not practical to calculate $E(I(\mathbf{x}))$ directly. Instead, we calculate the *Sampled Expected Improvement* (SEI) as suggested in Chipman et al. [8] and Chen et al. [6], i.e., to estimate $E(I(\mathbf{x}))$ based on the posterior samples of y,

$$\hat{E}(I(\mathbf{x})) = \sum_{m=1}^{M} (\max\{y^{(m)}(\mathbf{x}) - f_{\max}, 0\})/M, \tag{18}$$

where $y^{(m)}(\mathbf{x}) = f_N^{(m)}(\mathbf{x})$ is the mth posterior sample by (15), and M is the total number of posterior samples. Unlike in the Gaussian case, the SEI value in (18) may not be expressed as a weighted sum of the improvement term and the prediction uncertainty term. From its definition, only the prediction posterior samples $y^{(m)}(\mathbf{x})$ that are larger than the current best value, f_{max}, are taken in the summation. Thus SEI first identifies the possible "improvement" area, $\{\mathbf{x}|y^{(m)}(\mathbf{x}) > f_{max}$ for some $m\}$, and then sums over these terms.

A new explored point $\mathbf{x}_{N+1}$ at step $N+1$ is then selected to maximize the SEI criterion $\hat{E}(I(\mathbf{x}))$,

$$\mathbf{x}_{N+1} = \arg \max_{\mathbf{x} \in V \setminus P_{exp}} \hat{E}(I(\mathbf{x})), \tag{19}$$

where P_{exp} is the current explored point set.

3.3 The proposed algorithm and remarks

In the first part of this section, we will present a summary of the algorithm and the flexible usage of the proposed adaptive RBF-based global optimization framework. For abbreviation, we will refer to the proposed method as BaRBF, where "Ba" stands for "Bayesian adaptive". In the second part, we will compare our method with the baseline method proposed in Regis and Shoemaker [20].

One key element of the proposed BaRBF algorithm is to sequentially identify the next explored points. Since the SEI criterion does not have a closed form, it may not be easy to determine the next explored point by solving (19). Instead of directly solving (19) over $V \setminus P_{exp}$, we choose a set of candidate points, χ_N, from V first and then find the next point as

$$\mathbf{x}_{N+1} = \arg \max_{\mathbf{x} \in \chi_N} \hat{E}(I(\mathbf{x})). \tag{20}$$

There are two approaches for generating the candidate points.

- Pre-specify a grid over the experimental region, V, and treat the these unexplored grid points as candidates.
- Randomly and uniformly sample the candidate points from $V \setminus P_{exp}$.

For the scenario of pre-specified grid, this idea is quite natural. We simply specify a fixed grid, χ, over V before implementing the BaRBF, and set $\chi_N = \chi \setminus P_{exp}$. In practice, the precision of each variable should be limited and thus we can have the grid point set by setting the grid size as the variable precision. However, the problem for the grid set is the curse of dimensionality especially when the dimension optimization problem becomes larger. In addition, to keep a huge number of grid points in the process would slow down the computational speed. Instead of the grid point set, we may follow the idea in Regis and Shoemaker [20], i.e., we decide the number of candidate points before implementing the BaRBF, and then we uniformly sample the new candidate points from V to form χ_N at each iteration. In practice, given the current P_{exp}, we uniformly and independently sample the candidate points from V and once the selected points are in P_{exp}, we would replace the points by re-sampling them again.

Algorithm 1 summarizes the proposed global optimization method. In the beginning, the initial design is chosen as a space-filling design. In this paper, the maximin Latin hypercube design [17] is used. Then the main body of Algorithm 1 is to iterate between the two steps for the surrogate model construction in Sect. 3.1 and the point selection criterion in Sect. 3.2.

Note that the proposed BaRBF can be flexibly used in different scenarios. For example, when the number of available function evaluations is small to moderate, there may not be enough observations to estimate all the parameters. In this case, we only need to update some part of RBF parameters, say the scale parameter $\mathbf{s}$ by setting all the scale parameters $s_i \equiv s$, $(i = 1, \ldots, N)$, and need not update the $\boldsymbol{\mu}_i$ parameters. The choice of whether to update all parameters or part of them can be decided based on the magnitude of the model residuals at the initial stage. If updating all parameters leads to relative large model residuals, then

Algorithm 1 Global Optimization Algorithm

1: Choose a small set of initial explored points $P_{exp} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots . \mathbf{x}_{N_{min}}\}$ using a maximin Latin hypercube design, and evaluate $f(\mathbf{x}_i)$ on P_{exp}
2: **for** $N = N_{\min}, \ldots, N_{\max}$ **do**
3: Construct a Bayesian surrogate model as in Sect. 3.1 based on $\{(\mathbf{x}_i, f(\mathbf{x}_i)), i = 1, \ldots, N\}$
4: Generate the candidate point set, χ_N.
5: Calculate the SEI in (18), and select a new explored point based on (20).
6: Update $P_{exp} = P_{exp} \cup \mathbf{x}_{N+1}$ and evaluate $f(\mathbf{x}_{N+1})$
7: **end for**
8: **Return** the current best optimal point, $\hat{\mathbf{x}}_{\text{opt}} = \arg\max_{\mathbf{x} \in \mathbf{P_{exp}}} f(\mathbf{x})$ and the corresponding function value, $f(\hat{\mathbf{x}}_{\text{opt}})$.

we can fix certain parameters instead. The formulas of the posterior distribution in (8)–(14) need some minor changes accordingly if certain RBF parameters are fixed. For the above example, one only needs to set s_i in Eq. (8) to be the same s, and update only one s in (14), and does not need to update the $\boldsymbol{\mu}_i$'s in (12) and (13).

Now we consider the convergence property of Algorithm 1. Suppose the candidate set is based on a pre-specified grid. If we have enough resource, then we would be able to check all grid points and identify the best point over this grid. When we generate candidates from the uniform distribution over $V \backslash P_{exp}$, the convergence result in Theorem 1 of Regis and Shoemaker [20] is applicable to our situation. In that theorem, there are two important conditions for the generation of the candidate points. The first one is that the candidate points are conditionally independent given the current explored points. Since we independently generate the candidate points over $V \backslash P_{exp}$, this condition is satisfied in our approach. The second one, related to the generation distribution, is that there must be positive probability such that each candidate point falls within a δ-neighborhood of a point of V. In fact, this second condition is to ensure that every point has a positive probability to be selected as a candidate point. Following Regis and Shoemaker [20], our second approach does satisfy these two conditions because we generate candidates uniformly and randomly over V. Therefor, according to their Theorem 1, under certain conditions for the objective functions, the unique global optimal point in V can be identified almost surely.

For the remaining part of this section, we will compare our BaRBF with the Global metric stochastic RBF (G-MSRBF) algorithm proposed by Regis and Shoemaker [20] from a theoretical perspective. The G-MSRBF method will be regarded as the baseline method from now on. First we give a brief review. The G-MSRBF employs a surrogate model $S_N(\mathbf{x})$ using RBFs,

$$S_N(\mathbf{x}) = \sum_{i=1}^{N} \lambda_i r(\mathbf{x}; \mathbf{x}_i, s) + p(\mathbf{x}), \tag{21}$$

where $p(\mathbf{x})$ is a polynomial term. The RBF parameters in (21) are pre-specified, i.e., the RBF centers are set at the explored points $\mathbf{x}_i$, and s is pre-calculated at the initial stage of optimization. The model coefficients λ_i in (21) are estimated by solving a deterministic linear system of equation $\Phi\lambda = F$, where $\Phi_{ij} = r(\mathbf{x}_i; \mathbf{x}_j, s)$, $F = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N))^T$. And their point selection criterion

$$W_N(\mathbf{x}) = (1 - \omega_N^G) V_N^R(\mathbf{x}) + \omega_N^G V_N^D(\mathbf{x}). \tag{22}$$

is a weighted average of the scaled response prediction $V_N^R(\mathbf{x})$ with

$$V_N^R(\mathbf{x}) = \begin{cases} (S_N(\mathbf{x}) - S_N^{\min})/(S_N^{\max} - S_N^{\min}) \text{ for } S_N^{\max} \neq S_N^{\min}, \\ 1 \text{ o.w,} \end{cases} \tag{23}$$

and the maximin distance criterion $V_N^D(\mathbf{x})$ with

$$V_N^D(\mathbf{x}) = (d_N(\mathbf{x}) - d_N^{\min})/(d_N^{\max} - d_N^{\min}), \tag{24}$$

where $S_N^{\max} = \max\{S_N(\mathbf{x})\}$, $S_N^{\min} = \min\{S_N(\mathbf{x})\}$, $d_N(\mathbf{x}) = \min_{1 \le i \le N} ||\mathbf{x} - \mathbf{x}_i||^2$, $d_N^{\min} = \min d_N(\mathbf{x})$, $d_N^{\max} = \max d_N(\mathbf{x})$. The ω_N^G can take values in $\{1, 0.8, 0.6, 0.4, 0.2\}$ periodically. For example, if at time $N = 20$, $\omega_N^G = 0.8$, then at the next time $N = 21$, $\omega_N^G = 0.6$. Then a new point $\mathbf{x}_{N+1}$ is selected to maximize $W_N(\mathbf{x})$, and finally the global maximizer is also estimated by $\hat{\mathbf{x}}_{opt} = \arg\max_i f(\mathbf{x}_i)$.

Although both methods use RBFs, there are two main differences. First, the surrogate model is different. BaRBF uses a Bayesian surrogate model that provides not only predictions but also its uncertainties, while the G-MSRBF utilizes a deterministic surrogate model that only provides predictions. Because our proposed surrogate model is similar to ridge regression, the approximation of response is more robust and smooth compared to the interpolation surrogate model in G-MSRBF. The second difference lies in the choice of the selection criterion for new explored points. In our method, we utilize the expected improvement criterion $E(\max\{y - f_{\max}, 0\})$, which can be regarded as a *soft-thresholding* version of $E(y)$. As previously discussed, thresholding the prediction makes it easier to identify global optima. In addition, under the Gaussian prediction, EI criterion contains the measure of the prediction improvement and the model uncertainty. Here the part of the prediction improvement can be treated as exploitation and the uncertainty part is used to explore the search space. In G-MSRBF, the global exploration is based on the maximin distance criterion, $V_N^D(\mathbf{x})$, and the $V_N^R(\mathbf{x})$ is used for local refining. The weighted average of these two criteria is adopted with pre-specific weight pattern. Simulation studies will be presented in Sects. 4 and 5 to further understand and compare the empirical performance of the two methods.

4 Simulation study

To assess the performance of BaRBF, we compare it with G-MSRBF, which is regarded as the baseline method. To make a fair comparison, we center the response first and set the polynomial term in G-MSRBF $p(\mathbf{x})$ as 0. In Sect. 4.1, the candidate points are fixed as a pre-specified grid, χ, in the experimental region and both methods will be implemented over the same grid. Then in Sect. 4.2, both methods are implemented by randomly and uniformly generating the candidates over the experimental region, V.

In addition to the G-MSRBF, we also consider another global optimization approach based on Gaussian process surrogate model for comparisons. Jones et al. [13] proposed the efficient global optimization (EGO) approach by using the Gaussian process for surrogate construction. EGO starts from an initial point set. After evaluating the response values of the initial design points, a numerical optimization approach, like genetic algorithm, is used to obtain the MLE of the parameters in the Gaussian process and then the corresponding surrogate model is obtained. Since the Gaussian process prediction follows a normal distribution, the EI criterion in Eq. (17) is used to identify the next explored point over the feasible candidate point set, χ_N, and then the surrogate model is updated. Iterate these two steps until a stopping

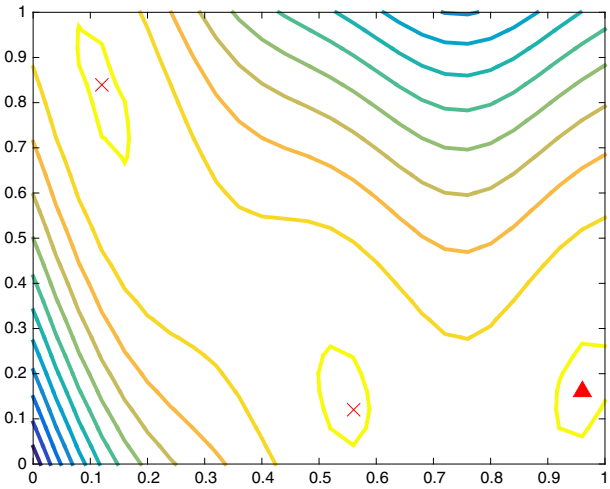

Fig. 1 The contour plot of the Branin function on $[0, 1]^2$ with grid size 0.04. The red triangle represents the global optimum and two red crosses denote the other two local optima. (Color figure online)

criterion is met. Usually the stopping criterion can be the number of explored points or the maximum value of the EI criterion over the unexplored points.

4.1 Grid candidate set

In this subsection, we demonstrate the performance of the BaRBF with a pre-specified grid set, and we refer it as grid BaRBF. First we start with a 2D global optimization example, whose objective function has few local optima. Then we consider another 2D objective function which is not smooth and has multiple global optima. Finally the higher-dimensional cases are also illustrated.

4.1.1 2D Branin function

We consider the standard 2D test function "Branin function", which has been widely used in the global optimization literature, e.g. Jones et al. [13]. The scaled version of "Branin function" we use here is defined as follows,

$$f(\mathbf{x}) = \frac{-1}{51.95}[(\bar{x}_2 - \frac{5.1\bar{x}_1^2}{4\pi^2} + \frac{5\bar{x}_1}{\pi} - 6)^2 + \left(10 - \frac{10}{8\pi}\right)\cos(\bar{x}_1) - 44.81], \qquad (25)$$

where $\bar{x}_1 = 15x_1 - 5$, $\bar{x}_2 = 15x_2$, and $x_1 \in [0, 1]$, $x_2 \in [0, 1]$. To simplify our code, we further restrict this function on the evenly spaced grid $\chi = [0, 0.04, \ldots, 1]^2$. The contour plot of the Branin function over the pre-specified grid points is given in Fig. 1, where there will be two local maxima and one global maximum on $[0.96, 0.16]$ with the maximum value 1.0473. In BaRBF, we first measure the prediction uncertainties for all grid points in χ and then identify the next explored point via (19) from the set $\chi \setminus P_{exp}$.

The objective is to find $\mathbf{x}$ that maximizes $f(\mathbf{x})$ in (25) with as few evaluations as possible. At each iteration of the algorithm, the current optimal point, $\hat{\mathbf{x}}_{opt}$, and its function value $f(\hat{\mathbf{x}}_{opt})$ are recorded together with all the explored points. We randomly choose a small set

of $N_{min}(= 16)$ initial explored points using a maximin Latin hypercube design [22]. All three methods start with the same set of $\mathbf{x}_i$'s. Each time the surrogate model is updated by incorporating the f value of a new explored point. Then we calculate and update the $\hat{\mathbf{x}}_{opt}$ value. For each algorithm, new explored points are selected and evaluated sequentially until the total number of explored points reaches $N_{max}(= N_{min} + 30) = 46$. This process is repeated 60 times, and the average performances are reported and compared for the two methods.

For fair comparison among BaRBF and G-MSRBF, we set the initial sampler of the RBF parameters in BaRBF to be the same as the fixed RBF parameters in G-MSRBF. Specifically, we use Algorithm 1 in Fasshauer and Zhang [9] to select an optimal value of s in G-MSRBF that minimizes a cost function that collects the errors for a sequence of partial fits to the data. The center parameters $\boldsymbol{\mu}_i$'s are set as the explored points $\mathbf{x}_i$'s.

From many simulation trials, we found out that, for the Branin test function, updating all parameters in BaRBF will lead to relatively large model residuals that do not converge. This might be caused by the small number of function evaluations. Thus we only update one scale parameter s with all $s_i \equiv s$ and fix the center parameter μ_i's at the explored points. We iterate the MCMC 10,000 times. Also, we discard the first 40% of the samples, and take 1 out of every 5 samples in the remaining 60% of the samples, in order to obtain stable and less correlated posterior samples for model fitting. In order to implement BaRBF, two important tuning parameters need to be pre-specified. The first one is the value of C in the coefficient prior. Here we set $C = 25$.

In this subsection, we first illustrate the proposed BaRBF with one particular simulation sample. Figure 2 plots the contours of the surrogate model in BaRBF and the locations of the explored points using BaRBF with $N = 16, 21, 26, 31, 36, 41$ for one simulation sample. Figure 2a shows the initial status of BaRBF. The initial design is a 16-run maximin Latin hypercube indicated by 16 green squares. The next explored point chosen by the selection criterion, i.e. the 17th point, is indicated by the black circle in the lower right corner. In Fig. 2b, the five additional points (17th to 21st) are indicated by the five blue squares.

These five points are divided into three sets, one closer to the global maximum, the other closer to the other two local maximums. As in Fig. 2a, the next explored point, i.e., the 22nd point, is indicated by the black circle. In Fig. 2c, all the 21 points from Fig. 2b are indicated by green squares, the additional five points by blue squares, and the next explored point by black circle. Then the same symbols are used in Fig. 2d–f to demonstrate the progression of points for $N = 31, 36$ and 41. Amazingly, except the initial design points, all the explored points are located closer to the three maxima, none for exploring bad regions. Finally the global maximum is identified in point 39 as shown by the black square in Fig. 2f. In summary these contour plots show that BaRBF efficiently explores the experimental space and quickly approaches the optimal points.

We report the performance of BaRBF and G-MSRBF based on 60 replications by randomly generating the initial LHD designs. The purpose is to see whether BaRBF provides a more efficient search path to identify the global maximum compared with G-MSRBF, for the same number of function evaluations. The numerical results are summarized in Table 1. First BaRBF has the higher mean value, 1.0443, than that of G-MSRBF, 1.0425, and is more stable because of its smaller standard deviation. Based on the sample quantiles of the optimal values identified by both approaches, there is a detectable difference at the 5% quantile values. We found out that for several cases, the best points identified by G-MSRBF are not close to the true optimal point and after checking the corresponding search processes, G-MSRBF did not efficiently explore the experimental space by properly choosing the next points. On the other hand, G-MSRBF performs better than BaRBF for the first quartile Q1. In addition,

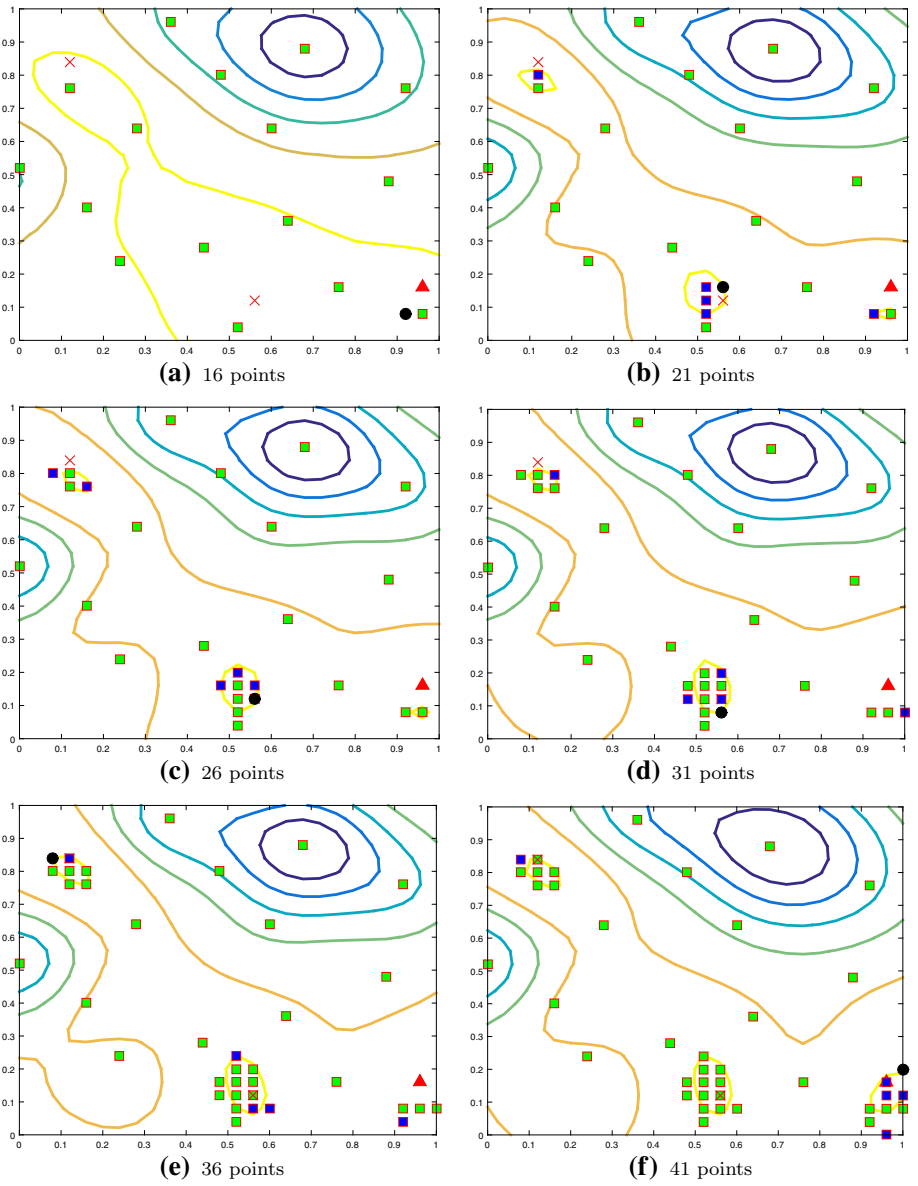

Fig. 2 The contours of surrogate model using grid BaRBF. Each of the six plots corresponds to a surrogate model with N = 16, 21, 26, 31, 36, 41 respectively (explanation of symbols is given in the text)

among the 60 replications, BaRBF can identify the true optimal values 26 times, which is higher than 20 times for G-MSRBF. Overall BaRBF has a better performance. Here we plot in Fig. 3 the mean value as well as the 5% and 95% quantile curves of the current optimal values with respect to the number of iterations for both methods. The mean curves in the two plots are very similar. More meaningful is the comparison of the two 5% quantile curves. For G-MSRBF, the curve moves up quickly until $N = 13$; then it gets stuck (flat) until about

Table 1 Summary of optimal values obtained by grid BaRBF, G-MSRBF and EGO with 60 replications in the 2-dimensional experiment with Branin function

Approach	5% Quantile	Q1	Median	Q3	95% Quantile	Mean	SD	Frequencies with true optimal values
BaRBF (SEI)	1.0397	1.0397	1.0464	1.0473	1.0473	1.0448	0.0033	29/60
G-MSRBF	1.0176	1.0438	1.0464	1.0473	1.0473	1.0425	0.0152	20/60
EGO	1.0473	1.0473	1.0473	1.0473	1.0473	1.0473	0.000	60/60

The run size of the initial design is 16, and the optimal value of the Branin function is 1.0473

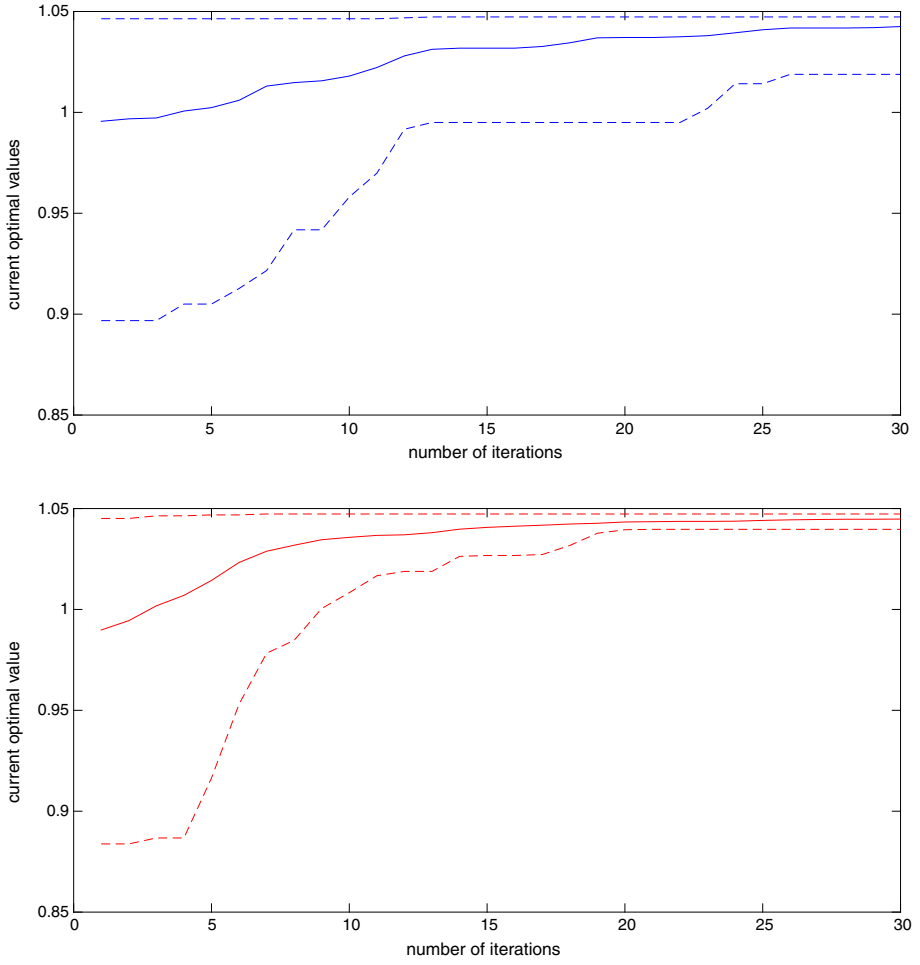

Fig. 3 The mean value (solid line) and the 5% and 95% quantiles (dashed line) of current optimal values based on 60 replications for the example of Branin function. Upper panel: G-MSRBF, lower panel: grid BaRBF

$N = 23$. By comparison, the 5% quantile curve for BaRBF moves up quickly until $N = 18$. By then, the band between the upper and lower quantile curves is very narrow and continues to shrink. The corresponding band for G-MSRBF does not shrink even to the end ($N = 30$). In fact it remains very wide when $N = 26$. This figure gives a more informative comparison than the numerical results in Table 1. It clearly shows the better performance of BaRBF over G-MSRBF.

When we compare the results with EGO, it seems that EGO works perfectly in this Branin example because EGO can quickly identify the global maximum point in each replications. The possible reason should be that since the Branin function can be treated as a smooth function, it can be fitted quit well by the Gaussian process model and thus the EI criterion in EGO can rapidly guide the search process to the target point. For our BaRBF, we do have a error assumption in the surrogate model and the model fitting may not be as well as Gaussian process model. In addition, SEI is computed as the sample expectation of the improvement function without any distributed assumption. Thus if the Gaussian assumption is satisfied, it is not surprised that EI can be more efficient in getting the global optimal point. In fact, when we monitor the search process of BaRBF, sometimes it may stay in a local area for a while. This should be related to that the exploration effect of the SEI criterion does not function well.

4.1.2 2D Ronkkonen function

In addition, we consider another 2D objective function in Rönkkönen et al. [21], i.e.,

$$f(x_1, x_2) = -\frac{1}{4}\sum_{i=1}^{2}[\cos(4\pi w_i) + 0.8\cos(8\pi w_i)], \tag{26}$$

where $w_i = \sum_{j=0}^{n_i}\binom{n_i}{j}P_{ij}(1-x_i)^{n_i-j}x_i^j$ for $i = 1, 2$, $n_1 = n_2 = 4$, and $P_1 = (0, 0.1, 0.2, 0.5, 1)$, $P_2 = (0, 0.5, 0.8, 0.9, 1)$ and the experimental region is $[0, 1]^2$. This objective function has been used as a test function in Chipman et al. [8]. Here we also restrict the function on the evenly spaced grid $\chi = [0, 0.04, \ldots, 1]^2$. The contour plot of this Ronkkonen function over the pre-specified grid points is given in Fig. 4, where there are 12 local maximums and 4 global maximal points with the maximum value, 0.4777. Because of its multiple local and global optimal points, this Ronkkonen function is not as smooth as the Branin function.

In this example, the initial point sets are the same as those in Sect. 4.1, and the total number of explored points is set as $N_{max} = 16 + 30 = 46$, i.e., based on 16 initial points, the search algorithm iterates 30 times by sequentially adding 30 points. Then the best value among the 46 explored points is reported. Here the goal is to identify one of the four global maximum points. The average performances of BaRBF, G-MSRBF and EGO over 60 replications are summarized in Table 2.

First, G-MSRBF performs worst in terms of the frequencies of reaching one of the four global maximum points (see the last column of Table 2). Then we focus on comparing the performance between BaRBF and EGO. From Table 2, the mean of the best function value found by BaRBF is 0.4775 with standard deviation 4.6850e−4, while the corresponding values for EGO are 0.4526 and 0.0344 respectively. We also compute the sample quantiles of the best values found by both methods. Table 2 shows that BaRBF touches the global maximum at the 50% sample quantile, while EGO reaches 0.4777 at the 75% quantile. In addition, for the BaRBF, the frequency of reaching a global optimum is 43/60, while the

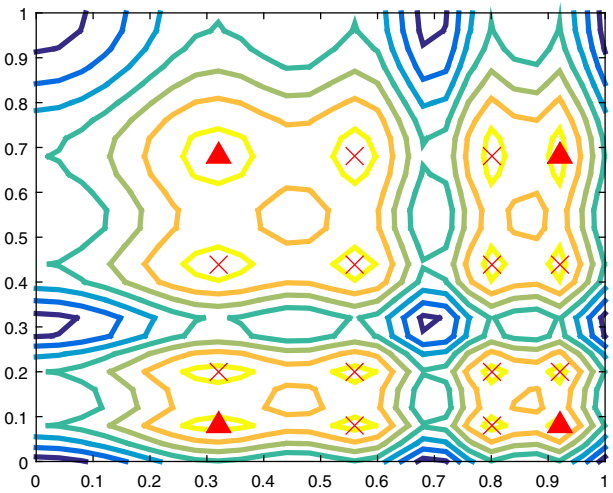

Fig. 4 The contour plot of the Ronkkonen function on $[0, 1]^2$ with grid size 0.04. The red triangle represents the global optimum and 12 red crosses denote the other two local optima. (Color figure online)

Table 2 Summary of optimal values obtained by BaRBF, G-MSRBF and EGO with 60 replications of the Ronkkonen function over a pre-spcified grid

Approach	5% Quantile	Q1	Median	Q3	95% Quantile	Mean	SD	Frequencies with true optimal values
BaRBF (SEI)	0.4766	0.4775	0.4777	0.4777	0.4777	0.4775	4.6850e−4	43/60
G-MSRBF	0.3635	0.4407	0.4766	0.4775	0.4777	0.4529	0.0363	11/60
EGO	0.3922	0.4405	0.4766	0.4777	0.4777	0.4526	0.0344	18/60

The run size of the initial design is 16, and the optimal value of the Ronkkonen function is 0.4777

frequency for the EGO is 18/60. The mean value of 60 replicates, and the 5% and 95% quantile curves of the current optimal values with respect to the number of iterations for EGO and BaRBF are shown in Fig. 5. The mean curve of the BaRBF moves up quickly to the global optimal value, while the curve for EGO moves up more slowly. In addition, the 5% quantile curve for EGO does not get much improvement before adding 25 explored points, i.e., $N = 16 + 25 = 41$, while the same curve for BaRBF moves up quickly and gets close to the global optimal value after adding 15 points, i.e., $N = 16 + 15 = 31$. Another attractive feature for BaRBF is the extremely low standard deviation (see the SD column), which may suggest that the BaRBF can perform stably over repeated implementations. In summary, the BaRBF outperforms the EGO in this example.

Chipman et al. [8] have pointed that when the test function has multiple global optima, the EI criterion might lead the search process to jump out of the neighborhood of one global optimum to another one and cannot stick around one global optimum. This may be explained by the fact that the EI criterion tries to minimize the prediction uncertainties among different global optima. The 5% quantile curve for EGO in Fig. 5 seems to support this point. For the BaRBF, by using SEI, it can quickly locate one neighborhood of a global maximum and then identify the best value. In addition, since this Ronkkonen function has more local optima than that of the Branin function and is more oscillating, the normality assumption and the

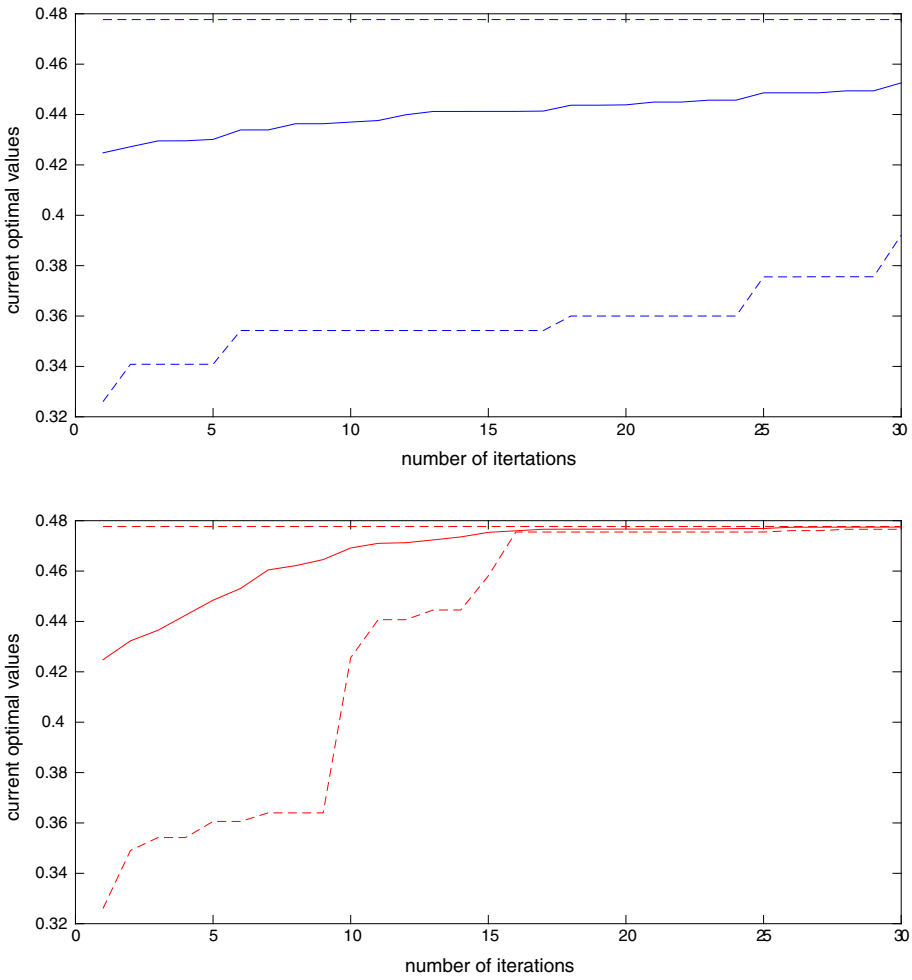

Fig. 5 The mean value (solid line) and the 5% and 95% quantiles (dashed line) of current optimal values based on 60 replications, Ronkkonen function. Upper panel: EGO, lower panel: grid BaRBF

interpolation property of the Gaussian process may not give advantages for the surrogate construction. For the BaRBF, the surrogate model consists of additive radial basis functions and their parameters are simultaneously adjusted via the proposed Bayesian approach. Thus our surrogate construction approach may be more advantageous for non-smooth test functions.

4.1.3 Simulation studies for three and four dimensions

In addition to these two 2D test functions, we consider 3D and 4D examples to illustrate that the grid BaRBF can deal with higher dimension problems and also compare its performances with EGO and G-MSRBF.

Two test functions are considered. The first one is the 3D Ronkkonen function [21] shown below,

$$f(x_1, x_2, x_3) = -\frac{1}{4}\sum_{i=1}^{3}[\cos(4\pi w_i) + 0.8\cos(8\pi w_i)], \tag{27}$$

where $w_i = \sum_{j=0}^{n_i}\binom{n_i}{j}P_{ij}(1-x_i)^{n_i-j}x_i^j$ for $i = 1, 2, 3$, $n_1 = n_2 = 4$, and $P_1 = (0, 0.1, 0.2, 0.5, 1)$; $P_2 = (0, 0.5, 0.8, 0.9, 1)$; $P_3 = (0, 0.6, 0.7, 0.9, 1)$. The second one is a 4D Hartmann function [19] defined as

$$f(x_1, x_2, x_3, x_4) = -\frac{1}{0.839}\left[1.1 - \sum_{i=1}^{4}\alpha_i \exp\left(-\sum_{j=1}^{4}A_{ij}(x_j - P_{ij})^2\right)\right], \tag{28}$$

where $\alpha = (\alpha_i) = (1.0, 1.2, 3.0, 3.2)$; $A = (A_{ij}) = \begin{pmatrix} 10 & 3 & 17 & 3.5 \\ 0.05 & 10 & 17 & 0.1 \\ 3 & 3.5 & 1.7 & 10 \\ 17 & 8 & 0.05 & 10 \end{pmatrix}$ and $P = (P_{ij}) = 10^{-4}\begin{pmatrix} 1312 & 1696 & 5569 & 124 \\ 2329 & 4135 & 8307 & 3736 \\ 2348 & 1451 & 3522 & 2883 \\ 4047 & 8828 & 8732 & 5743 \end{pmatrix}$. The experimental region considered is $[0, 1]^d$ with $d = 3$ and 4 respectively. According to Rönkkönen et al. [21], there are 5^3 local maximum points and 27 of them are the global maximum points. For the 4D Hartman function, there are fewer local optimal points.

The candidate set is based on a pre-specified grid. For the 3D case, we divide the experimental region into $(26)^3$ grid points by setting the grid size as 0.04. For this grid, there is only one global maximum point $(0.32, 0.68, 0.44)$ with the function value 0.3584. For the 4D case, we divide the experimental region into $(21)^4$ grid points by setting the grid size as 0.05. The maximum function value over these grid points is 3.1218. To implement the BaRBF, we follow the same RBF set-up in Sect. 4.1. That is, we set the scale parameter s with all $s_i \equiv s$ and fix the center parameter μ_i's at the explored points. For the tuning variable C, we fix C as 15 and 10 for $d = 3$ and 4 respectively. In both cases, the number of initial points and iterations are 50, and the initial points are from a maximim LHD with respect to the corresponding dimenaionality. Here the performance of the BaRBF is measured by the maximum function value identified among 100 explored points and is summarized based on 20 replications by independently re-generating the initial design points. For the comparison purpose, we also implement EGO and G-MSRBF.

Table 3 gives a summary of the maximum values obtained by the three approaches. First, consider the 3D case. G-MSRBF performs worst in this case because G-MSRBF cannot identify any true global maximum value within the 20 replications. Between BaRBF and EGO, BaRBF has better performance in the average maximum function values and the frequency of reaching the global maximum point, but both approaches share similar values in the first and third quartiles, Q1 and Q3, and the median value. Because there are few replications, the maximum value identified by EGO is less than 0.34. A possible reason should be similar to what was stated in Sect. 4.1.2, namely, the Ronkkonen function contain too many local optimal points and EGO may jump around different local modes. Finally, the SD value for BaRBF is much lower than that for the others. This is similar to what we observe in Table 2 for the 2D case and has a similar implication on the stability of the BaRBF. For this 4-dimensional Hartmann function, BaRBF outperforms G-MSRBF in terms of the frequency,

Table 3 Summary of maximum values obtained by BaRBF, G-MSRBF and EGO with 20 replications in the 3- and 4-dimensional cases with the grid candidate sets

Function	Approach	Q1	Median	Q3	Mean	SD	Frequencies with true optimal values
3D Ronkkonen fn.	BaRBF	0.3578	0.3580	0.3584	0.3581	3.3973e−04	9/20
	G-MSRBF	0.3203	0.3362	0.3576	0.3334	0.0234	0/20
	EGO	0.3579	0.3580	0.3583	0.3566	0.0048	5/20
4D Hartman function	BaRBF	3.1119	3.1218	3.1218	3.0936	0.0746	15/20
	G-MSRBF	3.0948	3.1218	3.1218	3.0903	0.0972	13/20
	EGO	3.1218	3.1218	3.1218	3.1099	0.0531	19/20

the mean of the optimal values and a smaller standard deviation. As shown in Table 3, for BaRBF, the middle 50% of values between Q1 and Q3 is extremely tiny and smaller than that for G-MSRBF. However, EGO has the best performance in this case. We think it should be related to the target function because this Hartmann function has few optimal points and thus it favors the EGO approach.

4.2 Uniform candidate points

In this subsection, we demonstrate the performance of the BaRBF with uniformly generating the candidate set, i.e., grid-free BaRBF. In addition to the four objective functions, four different benchmark problems for the special session and competition on Single Objective Real-Parameter Numerical Optimization in 2014 IEEE Congress on Evolutionary Computation (CEC) are considered [15]. The corresponding test functions with different dimensionalities, d, are shown in the following.

– High Conditioned Elliptic Function with $d = 2$ and 4:

$$f(\mathbf{x}) = -\sum_{i=1}^{d} (10^6)^{\frac{i-1}{d-1}} (2 \times (x_i - 0.5))^2.$$

– Ackley Function with $d = 2$, 4 and 6:

$$f(\mathbf{x}) = 20 \exp\left(-0.2 \sqrt{\frac{1}{d} \sum_{i=1}^{d} (2 \times (x_i - 0.5))^2}\right) \exp\left(\frac{1}{d} \sum_{i=1}^{d} \cos(2\pi (2 \times (x_i - 0.5))) - 20 - \exp(1)\right).$$

– Griewank Function with $d = 2$, 4 and 6:

$$f(\mathbf{x}) = -\sum_{i=1}^{d} \frac{100 \times (x_i - 0.5)^2}{4000} + \prod_{i=1}^{d} \cos\left(\frac{100 \times (x_i - 0.5)}{\sqrt{i}}\right) + 1.$$

– Rastrigin function with $d = 8$:

$$f(\mathbf{x}) = -10d - \sum_{i=1}^{d}[(x_i - 0.5) - 10\cos(2\pi(x_i - 0.5))].$$

Note that, following Liang et al. [15], we have modified these functions by shifting the center to be $(0.5, \ldots, 0.5)$ and re-scaling variables with different constants. For these 9 functions, when the experimental region is $V = [0, 1]^d$, with $d = 2, 3, 4, 6$, and 8, the global optimal point is $(0.5, \ldots, 0.5)$ with maximum value 0.Overall there are 13 test functions.

To implement grid-free BaRBF, G-MSRBF and EGO with respect to these test functions, we need to specify the following parameters. First, when the 2-; 3- and 4-dimensional problems are considered, the numbers of initial points and iterations are the same as in Sect. 4.1. For the 6-dimensional cases, we choose 50 initial points from a maximin LHD and iterate the approach 50 times. For the 8-dimensional case, there are 80 initial points from a maximin LHD and 60 iterations. For the grid-free approach, we need to choose certain numbers of candidates at each iteration. Here we uniformly sample $1000 \times d$ points from $V \setminus P_{exp}$. Consider the tuning parameters in BaRBF. We set $C = 15$ for the 2-, 6- and 8-dimensional cases and for the other dimensionalities, C is set as 25. Except for the value of C, the other parameters are the same as those in Sect. 3.1.3. For the 2D cases, we repeat the experiment 60 times; for the 3D; 4D and 6D cases, the number of the replication is 20. For the 8D case, we repeat 30 times. In addition, for each replication, we would regenerate the initial points from a maximin LHD independently. Finally, we collect the best values identified from the different approaches and summarize them as Tables 4 and 5. Use the result of the 8D Rastrigin function as an illustration. Figure 6 shows the corresponding 5%, 95% quantile curves and the mean values with respect to the number of iterations. Overall the results suggest that the proposed grid-free BaRBF does improve the objective function values over the whole search process; especially in the first few iterations, the improvement is significant. However, the improvement slows down later. The other cases share similar patterns.

Next we compare numerical results among three methods. Due to the different types of local optima, we cluster all functions into the three groups as follows.

Bowl-Shaped (Unimodal): 2D and 4D High Conditioned Elliptic Functions;
Few Local Optima: 2D Branin function and 4D Hartman function;
Many Local Optima: 2D, 4D, 6D Acklay functions; 2D, 4D, 6D Griewank functions; 2D, 3D Ronkkonen functions and 8D Rastrigin function.

We summarize the comparisons in the following:

Bowl-Shaped (Unimodal): In the high conditioned elliptic function for 2D and 4D, the range of response is large. From the the numerical results, BaRBF performs better than EGO and G-MSRBF, in terms of the quantile values, the mean values and the standard deviations. The EGO performs worst in the 4-dimensional case.

Few Local Optima: EGO and G-MSRBF outperform BaRBF in both cases. However, except for the standard deviations, the differences among the mean values and quantile values are small. In both case, EGO performs best in all measures.

Many Local Optima: BaRBF performs better in the cases of 2D Ackley function, 4D and 6D Griewank functions and 2D and 3D Ronkkonen functions. MSRBF is the best for the case of the 2D Griewank function. But EGO does better for the 4D and 6D Ackley function and the 8D Rastrigin function.

Overall, our approach, BaRBF, performs better in 7 out of 13 cases. Based on these numerical results, we have the following summary remarks:

Table 4 Summary of optimal values obtained by grid-free BaRBF, G-MSRBF and EGO with 60 replications in 2-dimensional cases

Function	Approach	5% Quantile	Q1	Median	Q3	95% Quantile	Mean	SD
2D elliptic function	BaRBF	− 0.1234	− 0.0495	− 0.0244	− 0.0102	− 0.0036	− 0.0336	0.0337
	G-MSRBF	− 1.0234	− 0.6464	− 0.4275	− 0.2223	− 0.0529	− 0.4674	0.3237
	EGO	− 0.2187	− 0.0809	− 0.0334	− 0.0153	− 0.0054	− 0.0615	0.0675
2D Branin function	BaRBF	1.0381	1.0448	1.0467	1.0471	1.0473	1.0454	0.0029
	G-MSRBF	1.0440	1.0457	1.0464	1.0471	1.0474	1.0461	0.0014
	EGO	1.0473	1.0474	1.0474	1.0474	1.0474	1.0474	4.0338e−5
2D Ronkkonen fn.	BaRBF	0.4737	0.4773	0.4778	0.4780	0.4781	0.4772	0.0018
	G-MSRBF	0.4723	0.4759	0.4773	0.4778	0.4781	0.4762	0.0033
	EGO	0.4154	0.4532	0.4735	0.4775	0.4777	0.4672	0.0209
2D Ackley function	BaRBF	− 0.0535	− 0.0255	− 0.0173	− 0.0104	− 0.0037	− 0.0212	0.0155
	G-MSRBF	− 0.0471	− 0.0316	− 0.0208	− 0.0138	− 0.0047	− 0.0232	0.0134
	EGO	− 0.1049	− 0.0744	− 0.0539	− 0.0384	− 0.0103	− 0.0575	0.0291
2D Griewank function	BaRBF	− 0.6324	− 0.3215	− 0.2026	− 0.1003	− 0.0386	− 0.2316	0.1668
	G-MSRBF	− 0.4115	− 0.2283	− 0.1507	− 0.1055	− 0.0357	− 0.1757	0.1127
	EGO	− 0.6067	− 0.4108	− 0.2675	− 0.1960	− 0.0890	− 0.2989	0.1563

Table 5 Summary of optimal values obtained by grid-free BaRBF, G-MSRBF and EGO in 3-; 4-; 6- and 8-dimensional cases

Function	Approach	Q1	Median	Q3	Max. value	Mean	SD
3D Ronkkonen fn.	BaRBF	0.3548	0.3566	0.3576	0.3582	0.3556	0.0031
	G-MSRBF	0.3203	0.3362	0.3576	0.3580	0.3334	0.0234
	EGO	0.3156	0.3363	0.3481	0.3562	0.3317	0.0198
4D elliptic function	BaRBF	− 12.4349	− 6.5470	− 3.4669	− 1.4580	− 8.1510	5.8892
	G-MSRBF	− 23.8559	− 15.3442	− 9.6406	− 6.2795	− 17.7994	10.4214
	EGO	− 2.5364*e*+3	− 1.2704*e*+3	− 134.225	− 16.3600	− 1.9430*e*+3	2.2748*e*+03
4D Hartman function	BaRBF	3.0832	3.1098	3.1151	3.1275	3.0781	0.0939
	G-MSRBF	3.1067	3.1151	3.1230	3.1280	3.1107	0.0150
	EGO	3.1118	3.1196	3.1245	3.1319	3.1181	0.0087
4D Ackley function	BaRBF	− 0.3541	− 0.2834	− 0.1638	− 0.0736	− 0.2904	0.2187
	G-MSRBF	− 0.2881	− 0.2366	− 0.1549	− 0.0764	− 0.2248	0.0814
	EGO	− 0.2133	− 0.1913	− 0.1529	− 0.0816	− 0.1899	0.0556
4D Griewank function	BaRBF	− 0.6437	− 0.4940	− 0.2500	− 0.0845	− 0.4503	0.2180
	G-MSRBF	− 0.7670	− 0.6489	− 0.4528	− 0.2664	− 0.6208	0.1885
	EGO	− 0.6774	− 0.6096	− 0.4396	− 0.1646	− 0.5744	0.2083
6D Ackley function	BaRBF	− 1.6341	− 1.4057	− 0.9976	− 0.5344	− 1.4184	0.5615
	G-MSRBF	− 1.1532	− 0.8149	− 0.6645	− 0.3591	− 0.9712	0.4709
	EGO	− 0.6017	− 0.5097	− 0.4625	− 0.2195	− 0.5177	0.1274
6D Griewank function	BaRBF	− 0.9265	− 0.7895	− 0.6924	− 0.3572	− 0.7775	0.1767
	G-MSRBF	− 0.9550	− 0.9205	− 0.7828	− 0.5892	− 0.8685	0.1374
	EGO	− 0.8994	− 0.8489	− 0.7285	− 0.5959	− 0.8216	0.1040
8D Rastrigin function	BaRBF	− 9.0985	− 6.5520	− 5.2748	− 3.5842	− 7.4002	2.9071
	G-MSRBF	− 9.4032	− 7.3887	− 5.9366	− 3.1462	− 7.6108	2.4987
	EGO	− 6.9078	− 6.1327	− 4.4958	− 2.6138	− 5.9096	1.7269

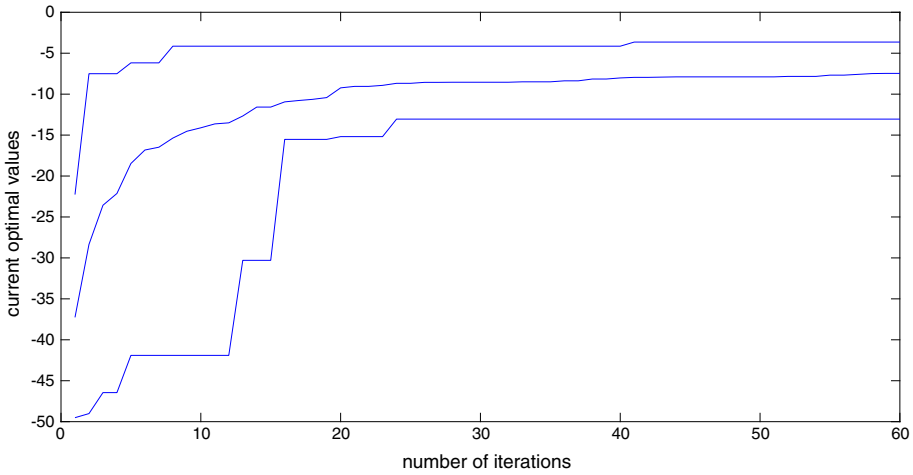

Fig. 6 Three lines are the 5% quantile; mean value and 95% quantile of current optimal values obtained by grid-free BaRBF based on 30 replications for the example of Rastrigin function

- Consider the cases of Branin function; 2D and 3D Ronkkonen functions and 4D Hartman function. The performances among the three methods share similar patterns whether the candidate points are randomly generated from a uniform distribution or based a pre-specified grid. For the Branin function and 4D Hartman function, EGO is the best approach. For the 2D and 3D Ronkkonen functions, BaRBF outperforms G-MSRBF and EGO.
- When the function contains few local optimal points, EGO can exploit its advantages due to its surrogate fitting.
- Based on our numerical results, our sequential approach would gear toward "exploitation". Thus we can do well for the cases with multiple local optimal points. Because part of the EI criterion is to select the next explored point to minimize the prediction variance of the surrogate model, it can be problematic for EGO when the objective function has several local optima. Instead of focusing on the search for a global optimal point, it can move around several local optimal points. A similar point was observed in Chipman et al. [8].
- A possible reason why EGO performs well for the 4D and 6D Ackley function is that there are not many local optimal points when we consider larger dimensionality.
- Based on the numerical results, G-MSRBF is usually the 2nd or 3rd best among the three method. For the cases of few local optima and many local optima, G-MSRBF share similar performances with that of EGO. However, G-MSRBF does not do well for the unimodal cases. This may be due to the choice of its selection criterion.

5 Discussions

In this session, several issues are studied. First we modify the proposed algorithm by adding a step to force the search process to jump out from a local optimal area. Then we study the effects of grid size on the performance of the proposed method when we choose the grid points as the candidate point set.

5.1 A modified version of BaRBF

From tracing the search process of the BaRBF in the Branin function example in Sect. 4.1, we found out that sometimes the BaRBF gets stuck in a local area and cannot leave the area for a while. In fact, even for the uniform candidates, the BaRBF can still get stuck in a local area. To overcome this potential weakness, we have the following modification. To jump out of this local area, we add an additional step, called the *escape step*, by monitoring the search process of the current best value. That is, we record the number of consecutive non-improvement iterations, i.e., iterations for which the current best function value cannot get improved from the new explored point. Denote this number by C_{non}. Once C_{non} exceeds a pre-specified number, M_I, it indicates that the BaRBF is stuck in a local area. Instead of continuing the search, we add some additional points to explore the experiment region. The additional point is chosen based on the maximin-distance criterion in the region. That is, given the current explored points, we find the point such that the union set of this point with the current explored points has the maximal value of the minimal distance between any two points in this union set. The purpose is to put additional points in the unexplored area as far away as possible from the existing points. We continue adding points until we obtain a better function value or until we add M_T points. Then we will return to the original search procedure. A similar idea has been adopted in Regis and Shoemaker [20] to detect if the search algorithm converges to a local optimum. We refer to this modification as M-BaRBF.

We implement the M-BaRBF for the Branin function by setting $M_I = M_T = 3$ by taking the pre-specified grid as the candidates, which is the same as shown in Sect. 4.1. For the same initial points sets and other tuning parameters, the average performance of the 60 replicates for M-BaRBF is shown in Table 6. Compare with the results for BaRBF in Table 1. Except for the 5% sample quantile value, the M-BaRBF performs better in the mean value and median value. In addition, the frequency to reach the global optimum is 38/60 which is much higher than 29/60 for the BaRBF. The Q1 value for M-BaRBF is significantly higher than that for BaRBF and the median value touches the global maximum of the Branin function.

5.2 The effects of the grid size

In the BaRBF, when we want to choose the next explored point from a grid set, we need to pre-specify the grid size. This size may be chosen based on the prior knowledge. In the numerical examples in Sect. 4.1, the grid size is fixed as 0.04 or 0.05. Suppose we can consider different grid sizes for a given optimization problem. We will illustrate the effects of the grid sizes on the performance.

We revisit the 2D Branin function example in Sect. 4.1. Instead of setting the grid size as 0.04, we choose the finer size 0.02 and divide the region into $(51)^2$ grid points which still cover the original grid, $[0, 0.04, \ldots, 1]^2$. Based on this finer grid, there are still two local maxima and the global optimal point is located at $[0.96, 0.16]$. Since the number of grid points is now about four times that of the original, we take more iterations, $4 \times 30 = 120$, for the proposed BaRBF. Then based on the same initial points and tuning parameters, the results with 60 replications are summarized in Table 7. In this table, in addition to the results with 120 iterations, we also report the results with 30 iterations for comparison purpose.

Compare the performances of BaRBF and BaRBF(120) in Tables 1 and 7 respectively. First, the BaRBF(120) is implemented over the finer grid and it has higher frequency, 40/60, to reach the global optimal point. In addition, the 5%, 25% quantiles and the median value of the best solutions of BaRBF(120) are 1.0471, 1.0471 and 1.0473 respectively which are

Table 6 Summary of optimal values obtained by M-BaRBF with Branin function

Approach	5% Quantile	Q1	Median	Q3 Quantile	95%	Mean	SD	Frequencies with true optimal values
M-BaRBF	1.0397	1.0464	1.0473	1.0473	1.0473	1.0458	0.0028	38/60

The run size of the initial design is 16, and the optimal value of the Branin function is 1.0473

Table 7 Summary of the 60 replications for the optimal values obtained by grid BaRBF with grid size, 0.02, in the 2-dimensional experiment with Branin function

Approach	5% Quantile	Q1	Median	Q3	95% Quantile	Mean	SD	Frequencies with true optimal values
BaRBF (120)	1.0471	1.0471	1.0473	1.0473	1.0473	1.0472	9.508e−5	40/60
BaRBF (30)	1.0466	1.0471	1.0471	1.0473	1.0473	1.0469	9.412e−4	17/60

The run size of the initial design is 16, and the optimal value of the Branin function is 1.0473

significantly higher than the corresponding values shown in Table 1. Obviously BaRBF(120) has the higher mean value 1.0472 and a smaller standard deviation. This may be related to the fact that there are more candidate points and larger number of iterations. Thus BaRBF can identify better function values due to finer grid and can still explore the experimental space because of a larger number of iterations. To support this guess, we also report the summary of the BaRBF with finer grid and 30 iterations, denoted by BaRBF(30). The corresponding 5%, 25% sample quantiles and the median value are still better than the corresponding values shown in Table 1. But the frequency for obtaining the global optimal point is only 17/60. It means that 30 iterations may not be large enough for BaRBF to explore the whole region and the search process may get stuck in some local areas. Thus we need to have more iterations to increase the probability to jump out of these local areas. Overall we can conclude that when we have a finer grid, a larger number of iterations should be necessary.

6 Conclusion and future work

We have proposed a global optimization framework that utilizes an adaptive RBF-based Bayesian surrogate model to approximate the true function, and to guide the selection of new points for function evaluation. There is novelty in both steps of the strategy. First, we use a hierarchical normal mixture surrogate model, where the parameters in the RBFs can be automatically updated to best approximate the true function. Second, the sample EI criterion is employed as a selection criterion. We have conducted some extensive numerical studies on standard test functions. The results demonstrate that the proposed BaRBF is more efficient and stable for searching the global maximizer compared with the G-MSRBF. For the comparison between the EGO and the BaRBF, their performance depends on the characteristics of the true objective functions. For example, when the true objective function has many local optimal points, like the 2D and 3D Ronkkonen functions, the BaRBF outperforms the EGO. Otherwise, the EGO perform better.

There are some directions for future research. First, the point selection criterion is a key element of BaRBF. A good selection criterion is to balance the trade-off between exploitation and exploration. Currently the Sampled EI criterion is adopted in the BaRBF. When the Gaussian prediction assumption is held, the EI criterion is a weighted sum of the prediction improvement and prediction variation which can be treated as a way to balance the exploitation and exploration. However, in the BaRBF, we do not have the distribution assumption and thus how SEI to balance these two properties is uncertain. Based on the numerical results in two 2D functions, our SEI criterion tends to have the exploitation property but less effect related to the exploration, because for the smooth Branin function, BaRBF may be stuck in a local area, but BaRBF can quickly identify the global maximum close to the explored points in the example of Ronkkonen function. Thus one possibility is to add the prediction variation measurement, i.e., to quantify the prediction uncertainties by the 95% confidence interval bandwidth of $f_N(\mathbf{x})$:

$$CIB(f_N(\mathbf{x})) = UCI(f_N(\mathbf{x})) - LCI(f_N(\mathbf{x})), \tag{29}$$

where $UCI(f_N(\mathbf{x}))$, $LCI(f_N(\mathbf{x}))$ are the upper CI and lower CI calculated as the 97.5% and 2.5% quantiles of the posterior samples $f_N^{(k)}(\mathbf{x})$. However, the problem should be how to integrate the SEI and CIB together.

Another issue is how to tune the proper parameters for BaRBF, especially the value of C. Revisit the Branin function example in Sect. 4.1. We did test the BaRBF with different

values of C like 5, 25, 100 and 150 by fixing the other parameters as done in Sect. 3.1.3. Overall, the performances of the optimal values are similar when $C > 5$. This supports our suggestion to set $C \geq 10$ and shows that the choice of C may not be too sensitive for the BaRBF. Of course, to identify the "best" C value is still problem-dependent. In addition, one future work is to have a data-driven tuning procedure for hyper-parameters in BaRBF. The tuning procedure suggested in Chen et al. [6] might serve as a starting point.

For the grid version of BaRBF, to identify the true optimal point, an adaptive grid BaRBF method can be considered as follows. At each iteration, we refine the current grid locally based on the hot spot areas identified from the surrogate surface, and then re-run grid BaRBF in these local areas independently. Take the Branin function example in Fig. 2 for illustration. In Fig. 2f, we can identify three hot spot areas. Then we can choose three smaller disjoint regions with finer grid to cover these three areas, and then individually implement BaRBF for each region. We can continue this procedure until the grid size in each region is small enough.

In this paper, grid-free BaRBF searches the whole experimental region V based on the surrogate model and then generates the candidates over V based on uniform distributed points. In some sense, the proposed approach can be treated as a global search method. In fact, Regis and Shoemaker [20] proposed a local MSRBF by sampling the candidates from a d-dimensional normal distribution centered at the current best point with a small variance. In order to cover the whole experimental region, local MSRBF can restart again and again until a stopping criterion is met. This approach is called the Multistart Local MSRBF (ML-MSRBF). In a similar fashion, the BaRBF can be modified accordingly. We can treat the area around the current best value as the local hot spot and restart the grid-free BaRBF by choosing candidates from a normal distribution centered at the best point in this local hot spot with a certain variance. After some number of iterations, we may identify another best point and restart the grid-free BaRBF again. Repeat this procedure until a stopping criterion is met. We leave it as a future work.

Acknowledgements Chen's research is supported by Ministry of Science and Technology (MOST) of Taiwan 104-2918-I-006-005 and the Mathematics Division of the National Center for Theoretical Sciences in Taiwan. Wu's research is supported by ARO W911NF-17-1-0007 and NSF DMS-1564438.

References

1. Andrieu, C., De Freitas, N., Doucet, A.: Robust full Bayesian learning for radial basis networks. Neural Comput. **13**(10), 2359–2407 (2001)
2. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
3. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, New York (2004)
4. Buhmann, M.D.: Radial Basis Functions: Theory and Implementations, vol. 12. Cambridge University Press, New York (2003)
5. Chen, R.B., Wang, W., Wu, C.F.J.: Building surrogates with overcomplete bases in computer experiments with applications to bistable laser diodes. IIE Trans. **43**(1), 39–53 (2011)
6. Chen, R.-B., Wang, W., Wu, C.F.J.: Sequential designs based on bayesian uncertainty quantification in sparse representation surrogate modeling. Technometrics **59**(2), 139–152 (2017)
7. Chipman, H., Hamada, M., Wu, C.F.J.: A Bayesian variable-selection approach for analyzing designed experiments with complex aliasing. Technometrics **39**(4), 372–381 (1997)
8. Chipman, H., Ranjan, P., Wang, W.: Sequential design for computer experiments with a flexible Bayesian additive model. Can. J. Stat. **40**, 663–678 (2012)
9. Fasshauer, G.E., Zhang, J.G.: On choosing "optimal" shape parameters for RBF approximation. Numer. Algorithms **45**(1–4), 345–368 (2007)
10. George, E.I., McCulloch, R.E.: Variable selection via Gibbs sampling. J. Am. Stat. Assoc. **88**(423), 881–889 (1993)

11. Gutmann, H.M.: A radial basis function method for global optimization. J. Glob. Opt. **19**(3), 201–227 (2001)
12. Jones, D.R.: A taxonomy of global optimization methods based on response surfaces. J. Glob. Opt. **21**(4), 345–383 (2001)
13. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive Black-Box functions. J. Glob. Opt. **13**(4), 455–492 (1998)
14. Koutsourelakis, P.S.: Accurate uncertainty quantification using inaccurate computational models. SIAM J. Sci. Comput. **31**(5), 3274–3300 (2009)
15. Liang, J., Qu, B., Suganthan, P.: roblem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. Technical report 201311 (2013)
16. Mockus, J., Tiesis, V., Zilinskas, A.: The application of Bayesian methods for seeking the extremum. Towards Glob. Optim. **2**, 117–129 (1978)
17. Morris, M.D., Mitchell, T.J.: Exploratory designs for computer experiment. J. Stat. Plan. Inference **43**, 381–402 (1995)
18. Oefelein, J.C., Yang, V.: Modeling high-pressure mixing and combustion processes in liquid rocket engines. J. Propul. Power **14**(5), 843–857 (1998)
19. Picheny, V., Wagner, T., Ginsbourger, D.: A benchmark of kriging-based infill criteria for noisy optimization. Struct. Multidiscip. Optim. **48**(3), 607–626 (2013)
20. Regis, R.G., Shoemaker, C.A.: A stochastic radial basis function method for the global optimization of expensive functions. INFORMS J. Comput. **19**(4), 497–509 (2007)
21. Rönkkönen, J., Li, X., Kyrki, V.: A framework for generating tunable test function for multimodal optimization. Soft. Comput. **15**, 1689–1706 (2011)
22. Santner, T.J., Williams, B.J., Notz, W.I.: The Design and Analysis of Computer Experiments. Springer, New York (2013)
23. Torn, A., Žilinskas, A.: Global Optimization. Springer, Berlin (1989)
24. Žilinskas, A.: On similarities between two models of global optimization: statistical models and radial basis functions. J. Glob. Opt. **48**, 173–182 (2010)
25. Žilinskas, A., Zhigljavsky, A.: Stochastic global optimization: a review on the occasion of 25 years of informatica. Informatica **27**, 229–256 (2016)
26. Zellner, A.: On assessing prior distributions and Bayesian regression analysis with g-prior distributions. Bayesian Inference Decis. Tech. Essays Honor Bruno De Finetti **6**, 233–243 (1986)