Global Optimization with Sparse and Local Gaussian Process Models

Tipaluck Krityakierne[1](✉) and David Ginsbourger[1,2]

[1] Department of Mathematics and Statistics, IMSV,
University of Bern, Bern, Switzerland
{tipaluck.krityakierne,ginsbourger}@stat.unibe.ch
[2] Idiap Research Institute, Martigny, Switzerland
ginsbourger@idiap.ch

Abstract. We present a novel surrogate model-based global optimization framework allowing a large number of function evaluations. The method, called SpLEGO, is based on a multi-scale expected improvement (EI) framework relying on both sparse and local Gaussian process (GP) models. First, a bi-objective approach relying on a global sparse GP model is used to determine potential next sampling regions. Local GP models are then constructed within each selected region. The method subsequently employs the standard expected improvement criterion to deal with the exploration-exploitation trade-off within selected local models, leading to a decision on where to perform the next function evaluation(s). The potential of our approach is demonstrated using the so-called Sparse Pseudo-input GP as a global model. The algorithm is tested on four benchmark problems, whose number of starting points ranges from 10^2 to 10^4. Our results show that SpLEGO is effective and capable of solving problems with large number of starting points, and it even provides significant advantages when compared with state-of-the-art EI algorithms.

Keywords: Black-box optimization · Expected improvement · Kriging

1 Introduction

In real world engineering optimization problems, the objective function is often a black box whose derivatives are unavailable, and function values are obtained from time-consuming simulations. To reduce the computational cost, in surrogate model-based optimization, the objective function is approximated with an inexpensive surrogate (also known as response surface model or metamodel). An auxiliary optimization problem on this surrogate is then solved in each iteration to determine at which point to evaluate the objective function next. The new data point is used to update the surrogate, and thus it is iteratively refined. Several popular response surface models such as radial basis functions, Gaussian process models (kriging), polynomials, and support vector regression have been successfully applied in this context (see, e.g. [5,8,10,13,14,19,23]).

P. Pardalos et al. (Eds.): MOD 2015, LNCS 9432, pp. 185–196, 2015.
DOI: 10.1007/978-3-319-27926-8_16

Thanks to its flexibility and efficiency, the EGO (Efficient Global Optimization) algorithm proposed by Jones [8] has become a very popular GP-based global optimization algorithm. It is based on the expected improvement criterion and more generally on ideas from Bayesian Optimization, following the seminal work carried out by Mockus and co-authors (see [9] and references therein). While EGO provides an elegant way to model the objective function and deal with the exploration versus exploitation trade-off, the computational cost and the storage requirements, nevertheless, have become major bottlenecks obstructing its practical application. Although quantifying complexity of EGO with hyperparameter re-estimation is a difficult task, EGO is known to be very slow and crash when the total number of observation points exceeds a few thousands. This is due to the training and prediction costs of GP that scale as $\mathcal{O}(N^3)$ and also the storage that scales as $\mathcal{O}(N^2)$, where N is the number of data points in the training set.

To circumvent this limitation, a number of sparse GP models have been proposed in the literature of GP regression (e.g. [3,15,16,21]). The idea behind these sparse models is generally to use a small number ($M << N$) of inducing points (also known as support points) to represent the full data points; as a result, the number of computations and storage requirements are reduced to $\mathcal{O}(NM^2)$ and $\mathcal{O}(NM)$, respectively. It is known that these approaches are related and can also be viewed within a single unifying framework. See [11] for details.

While some recent publications put a focus on Bayesian Optimization with a large number of points [18,22], to the best of our knowledge, no attempt has yet been made to integrate sparse GP within a global optimization framework. This work is intended as a contribution to the new area of applying GP-based global optimization to a larger number (typically, tens of thousands) of data points, which can be viewed somewhat as an extension to Bayesian Optimization.

In Sect. 2, we give necessary background regarding GP regression, Sparse Pseudo-input Gaussian Process models, as well as EI and EGO. In Sect. 3, we introduce the Sparse and Local EGO (SpLEGO) framework. A simple example of application on a one-dimensional problem and several numerical experiments that illustrate algorithm effectiveness in higher dimensions are presented in Sect. 4. Some comments on the proposed method as well as perspectives of future work are also given in this section. Finally, we conclude our work in Sect. 5.

2 Background

2.1 Problem Formulation and Notation

We consider a global optimization problem of the form:

$$\min_{\mathbf{x} \in \boldsymbol{D}} f\left(\mathbf{x}\right) \tag{1}$$

where $f : D \subset \mathbb{R}^d \to \mathbb{R}$ is assumed continuous and $D = [\mathbf{a}, \mathbf{b}] = \prod_{i=1}^{d} [a_i, b_i]$ ($a_i, b_i \in \mathbb{R} : a_i < b_i$). The objective function f is assumed to be expensive and without any derivative information available (referred to as "black box" henceforth).

The goal of this paper is to develop a GP-based global optimization algorithm that can find near globally optimal solutions when the number of starting points (or allowable function evaluations) is relatively large.

2.2 Gaussian Process Modeling

Suppose that we have observed the vector of outputs $f\left(\mathbf{X}\right)=\left[f\left(\mathbf{x}_1\right),...,\, f\left(\mathbf{x}_N\right)\right]^T$ at the training input points $\mathbf{X}=\left\{\mathbf{x}_1,...,\,\mathbf{x}_N\right\}$. Assume a Gaussian Process prior $f\sim \mathrm{GP}\left(\mu_0\left(\cdot\right),K\left(\cdot,\cdot\right)\right)$ where μ_0 and K are a given mean function and covariance kernel, respectively. For a fixed $\mathbf{x}\in D$, the posterior of $f\left(\mathbf{x}\right)$ knowing $f\left(\mathbf{X}\right)$ is $f\left(\mathbf{x}\right)|f\left(\mathbf{X}\right)\sim\mathcal{N}\left(\mu_N\left(\mathbf{x}\right),\sigma_N^2\left(\mathbf{x}\right)\right)$. Taking $\mu_0\left(\cdot\right)=0$, we have [12]

$$\mu_N\left(\mathbf{x}\right)=K\left(\mathbf{x},\mathbf{X}\right)K\left(\mathbf{X}\right)^{-1}f\left(\mathbf{X}\right) \tag{2a}$$

$$\sigma_N^2\left(\mathbf{x}\right)=K\left(\mathbf{x},\mathbf{x}\right)-K\left(\mathbf{x},\mathbf{X}\right)K\left(\mathbf{X}\right)^{-1}K\left(\mathbf{X},\mathbf{x}\right), \tag{2b}$$

where $K\left(\mathbf{X},\mathbf{x}\right)$ is defined as $\left[K\left(\mathbf{x}_1,\mathbf{x}\right),...,\,K\left(\mathbf{x}_N,\mathbf{x}\right)\right]^T$. $K\left(\mathbf{X}\right):=K\left(\mathbf{X},\mathbf{X}\right)$ (assumed invertible here) is defined analogously. One example (among many others, see [20]) of a commonly used covariance kernel is the squared exponential:

$$K\left(\mathbf{x},\mathbf{x}'\right)=\sigma^2\exp\left(-\frac{1}{2}\sum_{k=1}^{d}\theta_k\left(\mathbf{x}_k-\mathbf{x}'_k\right)^2\right), \tag{3}$$

where $\Psi=\left\{\sigma^2,\,\theta_1,...,\,\theta_d\right\}$ are the hyperparameters, whose values are often obtained by maximizing the log marginal likelihood:

$$\mathcal{L}\left(\Psi\right)=-\frac{1}{2}\log\left|K_\Psi\left(\mathbf{X}\right)\right|-\frac{1}{2}\mathbf{f}(\mathbf{X})^TK_\Psi^{-1}\left(\mathbf{X}\right)\mathbf{f}(\mathbf{X})-\frac{N}{2}\log\left(2\pi\right). \tag{4}$$

Evaluating any of Eqs. 2a, 2b or 4 relies on the inversion of the $N\times N$ covariance matrix $K_\Psi\left(\mathbf{X}\right)$, and so GP modelling is prohibitively expensive when the size of the training data set becomes large.

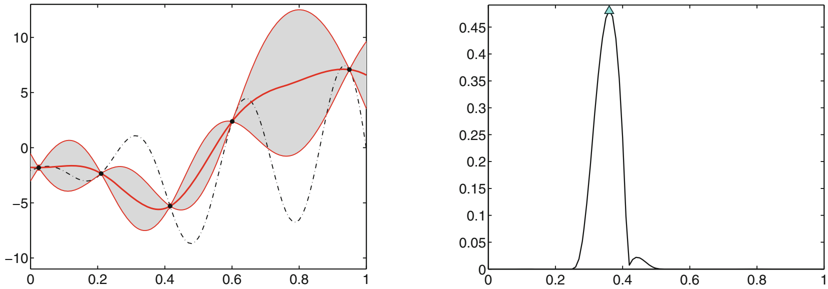
2.3 Expected Improvement and EGO

As in most surrogate-based optimization methods, EGO [8] starts by constructing a space-filling design in the decision space $\left\{\mathbf{x}_1,...,\,\mathbf{x}_{N_0}\right\}\subset D$, for some $N_0\geq 1$. The objective function is then evaluated at these design points and an initial GP model is fitted. The algorithm selects the next function evaluation point(s) by maximizing the expected improvement (EI) criterion, which depends both on the prediction $\mu_{N_0}\left(\mathbf{x}\right)$ and on the associated uncertainty $\sigma_{N_0}^2\left(\mathbf{x}\right)$ from Eqs. 2a and 2b.

More generally, for $N\geq N_0$, let $f_{\min}=\min\left\{f\left(\mathbf{x}_1\right),...,\,f\left(\mathbf{x}_N\right)\right\}$ be the current best objective function value. EI, defined as the expectation of the improvement brought by evaluating f at a candidate point, can be calculated analytically:

$$\mathrm{EI}_N(\mathbf{x}) = \mathbb{E}_N\left[\max\left(0, f_{\min} - f(\mathbf{x})\right)\right] \tag{5a}$$
$$= (f_{\min} - \mu_N(\mathbf{x}))\Phi\left(\tfrac{f_{\min} - \mu_N(\mathbf{x})}{\sigma_N(\mathbf{x})}\right) + \sigma_N(\mathbf{x})\phi\left(\tfrac{f_{\min} - \mu_N(\mathbf{x})}{\sigma_N(\mathbf{x})}\right), \tag{5b}$$

where $\mathbb{E}_N$ is the expectation taken with respect to posterior distribution given the first N observations, and Φ and ϕ are the standard Gaussian cdf and pdf, respectively. In EI algorithms such as EGO, at each iteration a function evaluation is performed at a point maximizing EI, i.e. $\mathbf{x}_{N+1} \in \mathrm{argmax}_{\mathbf{x}\in D}\,\mathrm{EI}_N(\mathbf{x})$, and the GP model is then updated with the new evaluation result. Figure 1 shows an example of GP model based on five observations (Left) and the corresponding EI criterion (Right). By assigning large values to inputs $\mathbf{x}$ whose $f(\mathbf{x})$ is likely to be less than $f_{\min}$ and/or whose prediction variance is high, the EI criterion provides a good balance between exploration of unexplored regions and exploitation of promising regions with low predictive mean.



(a) GP Predictive distribution (b) EI criterion

Fig. 1. A GP model and the corresponding expected improvement function. In Panel (a) the dots represent the training data. The black dashed line is the (unobserved) objective function. The shaded area represents the point-wise mean (middle red line) plus and minus twice the prediction standard deviation at each input value. In Panel (b), the point that attains the max EI is depicted by a blue triangle (Color figure online).

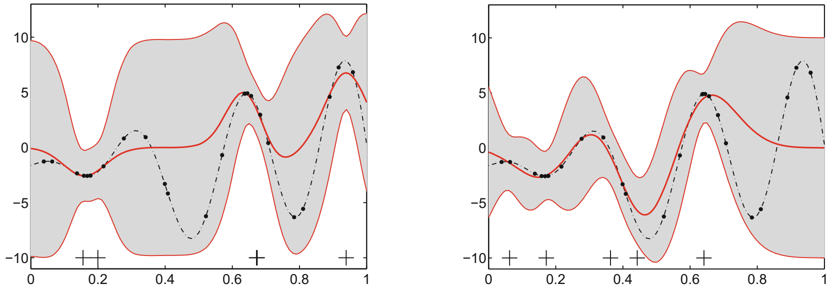
One remark regarding EI-optimal points shall be given before we turn to the next section. Let us consider the bi-objective optimization problem:

$$\min_{\mathbf{x}\in \boldsymbol{D}} F(\mathbf{x}) = \left(\mu_N(\mathbf{x}),\, -\sigma_N^2(\mathbf{x})\right). \tag{6}$$

Since EI is decreasing in $\mu_N(\cdot)$ and increasing in $\sigma_N(\cdot)$, if $\mathbf{x}$ maximizes the EI criterion then $\mathbf{x}$ is automatically in the Pareto set of the bi-objective problem above. In situations when the EI formula (Eq. 5b) is not applicable, using this Pareto optimality property instead can come in handy, as we will see in Sect. 3.

2.4 Sparse Pseudo-input Gaussian Process

To circumvent the time-complexity, storage bottlenecks, and potential singularity problems for a large covariance matrix, a number of computationally efficient sparse GP approximations have been proposed in the machine learning literature. In this section, we give a brief review of a particular method called Sparse Pseudo-input Gaussian Process (SPGP) [16]. The SPGP method is based on a low-rank approximation to the full GP covariance using a small set of inducing points $\bar{\mathbf{X}} = \{\bar{\mathbf{x}}_1, ..., \bar{\mathbf{x}}_M\}$. In SPGP, the inducing points are referred to as "pseudo-inputs" since they do not need to be a subset of the input training data but are rather inferred along with the kernel hyperparameters.



(a) Inducing points No. 1

(b) Inducing points No. 2

Fig. 2. SPGP predictive distribution obtained using different sets of inducing points

We first give two examples of SPGP predictive distribution using different sets of inducing points in Fig. 2. The black dots correspond to 25 training points. The locations of the inducing points are shown as crosses. The shaded areas represent the point-wise SPGP predictive means plus and minus twice the standard deviations at each input value. Again, the dashed black line represents the (unobserved) objective function. We can see that the predictive distribution is significantly influenced by the locations of the inducing points.

Coming to the SPGP equations, let us assume a zero mean GP prior on the objective function, $f \sim \mathrm{GP}\left(0, K\left(\cdot,\cdot\right)\right)$. Without going into the details of its derivation, it turns out that SPGP can be considered as a standard GP with a particular covariance function [16]:

$$K^S(\mathbf{x},\,\mathbf{x}') = Q(\mathbf{x},\,\mathbf{x}') + \delta_{\mathbf{x},\mathbf{x}'}\left[K(\mathbf{x},\,\mathbf{x}) - Q(\mathbf{x},\,\mathbf{x})\right], \tag{7}$$

where $Q(\mathbf{x},\,\mathbf{x}')=K\left(\mathbf{x},\bar{\mathbf{X}}\right)K\left(\bar{\mathbf{X}}\right)^{-1}K\left(\bar{\mathbf{X}},\mathbf{x}'\right)$ and $\delta_{\mathbf{x},\mathbf{x}'}$ is Kronecker's delta. After matrix simplifications, the predictive mean and variance of SPGP boil down to formula involving only calculations with matrices of manageable dimensionality:

$$\mu^S\left(\mathbf{x}\right) = K\left(\mathbf{x},\bar{\mathbf{X}}\right)H^{-1}K\left(\bar{\mathbf{X}},\mathbf{X}\right)\Lambda^{-1}f\left(\mathbf{X}\right) \tag{8a}$$

$$\sigma^{S2}\left(\mathbf{x}\right) = K\left(\mathbf{x},\mathbf{x}\right) - K\left(\mathbf{x},\bar{\mathbf{X}}\right)\left(K\left(\bar{\mathbf{X}}\right)^{-1} - H^{-1}\right)K\left(\bar{\mathbf{X}},\mathbf{x}\right), \tag{8b}$$

where $\Lambda = \mathrm{diag}\left(K\left(\mathbf{X}\right) - Q\left(\mathbf{X}\right)\right)$ and $H = K\left(\bar{\mathbf{X}}\right) + K\left(\bar{\mathbf{X}},\mathbf{X}\right)\Lambda^{-1}K\left(\mathbf{X},\bar{\mathbf{X}}\right)$. Consequently, the pseudo-inputs $\bar{\mathbf{X}}$ can be considered as extra hyperparameters of the model and can be estimated jointly with the kernel hyperparameters (of size $Md + |\Psi|$) by maximizing the log marginal likelihood as in Eq. 4. Note that since here $K^S_{\Psi,\bar{\mathbf{X}}}(\mathbf{X})$ can be written as a sum of a low rank part and a diagonal part, it can be inverted in $\mathcal{O}\left(NM^2\right)$. See [16] for more details.

3 Sparse and Local GP for Global Optimization

We wish to have an EGO-like algorithm that offers expected improvement but can also handle a large number of starting points. One natural extension of EGO in such a situation is to partition the whole decision space into smaller subregions, $D = \cup_{i=1}^r R_i$, e.g. where each region R_i would contain no more than k training input points. Local GP models could then be constructed, and a point $w_i \in R_i$ maximizing the local $\mathrm{EI}^{(i)}$ could be identified within each region R_i. Finally, one could take the best point among all w_i (with the largest local EI) as an approximation to the solution point corresponding to the true global EI.

While this may seem simple and appealing at first glance, such an approach would actually raise a few issues, and could be very computationally expensive in practice especially because of the potentially large number of local GP models to build and maintain. Our proposed algorithm, called SpLEGO (**Sp**arse and **L**ocal **EGO**), on the other hand, takes advantage of space partitioning while remaining at a more reasonable computational cost through some kind of pruning.

Given $\mathbf{X}_N = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ and $\mathbf{Y}_N = \{f(\mathbf{x}_1), ..., f(\mathbf{x}_N)\}$, the specific steps of SpLEGO are given in Algorithm 1.

Algorithm 1. SpLEGO Framework

```
1. Build a sparse GP model using $M << N$ inducing points $\mathcal{I}_{\mathrm{IP}} = \{\bar{\mathbf{x}}_1, ..., \bar{\mathbf{x}}_M\}$.
2. Identify center points for local models, $V = \{v_1, ..., v_r\}$:
   (a) Generate a Quasi-random sequence, e.g. Sobol sequence, $Q = \{u_1, ..., u_q\} \subset D$.
   (b) Compute the sparse predictive mean and variance for all $u \in Q$.
   (c) Identify the Pareto front with the two objectives, $F_1(\mathbf{x}) = \mu^S(\mathbf{x})$ and $F_2(\mathbf{x}) = -\sigma^{S2}(\mathbf{x})$. Let $v_1, ..., v_r \in Q$ be the points in the Pareto set.
3. For $i = 1 : r$,
   (a) Identify a subregion $R_i$ around $v_i$.
   (b) Build a local GP model.
   (c) Calculate local $\mathrm{EI}^{(i)}$. Let $w_i \in \mathrm{argmax}_{\mathbf{x} \in R_i} \mathrm{EI}^{(i)}(\mathbf{x})$, i.e. $w_i \in R_i$ maximizes the local $\mathrm{EI}^{(i)}$ using the local GP in Step 3b. Note that the global $f_{\min} = \min \mathbf{Y}_N$ is used as a threshold when calculating all local $\mathrm{EI}^{(i)}$'s.
4. Let $i_0 \in \mathrm{argmax}_{1 \le i \le r} \mathrm{EI}^{(i)}(w_i)$, $\mathbf{x}_{N+1} \leftarrow w_{i_0}$ and $y_{N+1} \leftarrow f(\mathbf{x}_{N+1})$.
5. Update $\mathbf{X}_{N+1} \leftarrow \mathbf{X}_N \cup \{\mathbf{x}_{N+1}\}$, $\mathbf{Y}_{N+1} \leftarrow \mathbf{Y}_N \cup \{y_{N+1}\}$, and $N \leftarrow N + 1$.
6. Go back to Step 1.
```

To grasp a big picture of the entire domain, SpLEGO first constructs a sparse global GP model (Step 1 of Algorithm 1). Following the same philosophy of EI criterion that favors regions with high uncertainty and low mean predictions, non-dominated points are then identified from a space filling sequence, where evaluations are done on the the two competing objectives (mean and variance) obtained from the sparse GP model (Step 2).

From the trade-off point-of-view, input points in a vicinity of Pareto-optimal points define interesting regions. Thereby, local GP models are built within each of these regions (Step 3b). Finally, the next evaluation point is taken as the point that attains the overall maximum local EI across all subregions (Step 4).

Steps 3a and b of Algorithm 1 need further clarification. While different approaches can be used to define a subregion R_i in Step 3a, in this work we define R_i to be a hyperrectangle $\left[\min\left(\mathbf{X}_N^{(i)} \cup \{v_i\}\right), \max\left(\mathbf{X}_N^{(i)} \cup \{v_i\}\right)\right]$, where $\mathbf{X}_N^{(i)} \subset \mathbf{X}_N$ is a set of k-nearest input neighbors of v_i, and the minimum and maximum are taken component-wise. Next, two possibilities of a local GP model in Step 3b are presented:

V1. Exact Local GP: Use the k points in $\mathbf{X}_N^{(i)}$ (with their corresponding exact observations) to build a local GP in the region R_i.

V2. Globalized Local GP: Use a combination of the k points in $\mathbf{X}_N^{(i)}$ and M noisy inducing points in $\mathcal{I}_{\text{IP}}$ from Step 1.

The details of the GP posterior used in version V2, which combines exact responses from the ith local model ($1 \leq i \leq r$) and noisy responses from the inducing points of the SPGP model, are as follows: Let $\mathbf{X}_N^{(i)} = \left\{\mathbf{x}_{N,1}^{(i)}, ... \mathbf{x}_{N,k}^{(i)}\right\}$ and $f\left(\mathbf{X}_N^{(i)}\right) = \left[f\left(\mathbf{x}_{N,1}^{(i)}\right), ..., f\left(\mathbf{x}_{N,k}^{(i)}\right)\right]^T$ be the exact k input-output observations from region i. Let $\bar{\mathbf{X}}_M = \{\bar{\mathbf{x}}_1, ... \bar{\mathbf{x}}_M\}$ be the (noisy) inducing points, with SPGP predictive means $\mu^S(\bar{\mathbf{X}}_M)$ and variances $\tau^2 = \sigma^{S2}(\bar{\mathbf{X}}_M)$. Writing $\tilde{\mathbf{X}}_N^{(i)} = \left[\mathbf{X}_N^{(i)}, \bar{\mathbf{X}}_M\right]$ and $\varDelta = \text{diag}(\tau^2)$, the predictive mean and variance of the combined local GP in region i are given by

$$\mu_N^{(i)}(\mathbf{x}) = K\left(\mathbf{x}, \tilde{\mathbf{X}}_N^{(i)}\right)\left[K\left(\tilde{\mathbf{X}}_N^{(i)}\right) + \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \varDelta \end{pmatrix}\right]^{-1} \begin{bmatrix} f\left(\mathbf{X}_N^{(i)}\right) \\ \mu^S(\bar{\mathbf{X}}_M) \end{bmatrix} \tag{9a}$$

$$\sigma_N^{(i)2}(\mathbf{x}) = K(\mathbf{x},\mathbf{x}) - K\left(\mathbf{x},\tilde{\mathbf{X}}_N^{(i)}\right)\left[K\left(\tilde{\mathbf{X}}_N^{(i)}\right) + \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \varDelta \end{pmatrix}\right]^{-1} K\left(\tilde{\mathbf{X}}_N^{(i)},\mathbf{x}\right). \tag{9b}$$

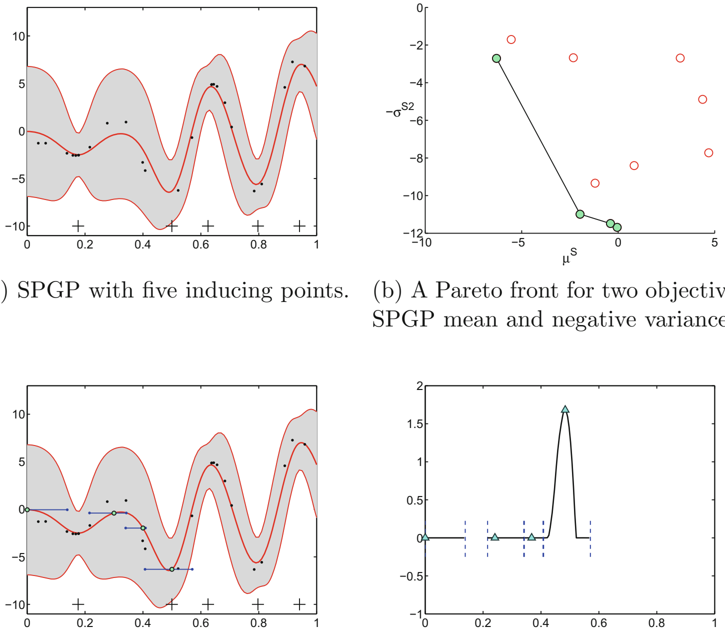
One can view Step 3b as a refinement phase. The difference between the two versions is the use of inducing points in SpLEGO-V2. While SpLEGO-V1 focuses on refining the selected regions of interest using only the exact evaluation points in the region R_i, SpLEGO-V2 uses information from both the nearby exact observations and the sparse global model.

Let us remark that although it seems out of reach here to specify the complexity of EGO or SpLEGO, the overall complexity of a typical step of the two algorithms are dominated by $\mathcal{O}(N^3)$ and $\max\left\{\mathcal{O}\left(NM^2\right), \mathcal{O}\left(k^3\right), \mathcal{O}\left(dN^2\right)\right\}$, respectively. Therefore, when $N \gg M$, k, d (which is our case), it appears that SpLEGO will be more efficient than EGO.

4 Applications

4.1 A Didactic Example

Figure 3 illustrates the application of SpLEGO with a simple didactic example. SPGP is used in Step 1 of the algorithm. Panels (a) and (b) correspond to Steps 1 and 2. Panel (c) corresponds to Step 3a. Here, four Pareto optimal points are used to define the same number of subregions R_i with $k = 3$ points each. Finally, Panel (d) corresponds to Step 3c where the local $\mathrm{EI}^{(i)}$'s are calculated within each R_i.



(a) SPGP with five inducing points.

(b) A Pareto front for two objectives: SPGP mean and negative variance.

(c) Pareto optimal points (green dots) and their subregions $\mathrm{R_i}$ (blue lines).

(d) Local EIs and points attaining their maxima (blue triangles).

Fig. 3. A step-by-step application with a simple didactic example (Color figure online)

4.2 Numerical Experiments

Test Problems. SpLEGO is assessed on four benchmark problems. The test functions have between 4 and 10 dimensions and are summarized in Table 1.

Table 1. Summary of test problems

Problem	Domain	N_1	N_2
Rastrigin [6]	$[-20, 20]^{10}$	100	0
Hartmann [4]	$[0, 1]^6$	400	5×200
Ackley [1]	$[-1, 3]^{10}$	280	4×280
Shekel [4]	$[0, 10]^4$	5000	1×5000

Initial Data. To examine method applicability, we create the initial data in a way that the points are packed in some regions but not completely filling the space. The initial designs are composed of two types of samples:

I1: Latin Hypercube Designs of size N_1
I2: clusters of uniformly distributed points of size N_2.

The total number of initial points is therefore $N_0 = N_1 + N_2$, where N_1, N_2 for each test problem are given in Table 1. For example, the initial design of Hartmann-6D consists of $N_0 = 1400$ points: a Latin Hypercube Design ($N_1 = 400$) and five clusters of 200 uniformly distributed points ($N_2 = 1000$).

Experimental Results. Ten trials are performed for both EGO and SpLEGO. Parameter values used in numerical experiments for SpLEGO are $M = 20$, $q = 500$, $k = 50$ (Steps 1, 2, and 3a of Algorithm 1). Here, we implement SpLEGO-V2. The plots of the average best objective function value ($\min_{1 \leq n \leq N} f(\mathbf{x}_n)$) versus number of sample size N (starting from N_0) are shown in Fig. 4. The top left panel of Fig. 4 corresponds to Rastrigin-10D. With a relatively small initial design of size $N_0 = N_1 = 100$ we did not expect SpLEGO to work that well; nevertheless, we see that our method outperforms EGO on this test problem. With a larger number of initial data, SpLEGO again outperforms EGO on Hartmann-6D and Ackley-10D (top right and bottom left panels). Finally, the bottom right panel illustrates the feasibility of using our method for very large N_0 (size 10^4) on Shekel-4D test function. Note that EGO is no longer feasible. For this test function, the results based on two versions of SpLEGO are shown.

Recall that while SpLEGO-V1 only relies on exact points from the neighbourhood R_i, SpLEGO-V2 incorporates furthermore the inducing points of SPGP when fitting local GPs. The algorithm achieves better results with SpLEGO-V2 for this example, note however that, in general the results may vary from problem to problem.

4.3 Comments and Perspectives of Future Work

Since SPGP relies on a set of inducing points (which is changed in every iteration), whether or not SPGP leads to regions containing the global minimum is still an open problem. Nevertheless, the presented results are promising for

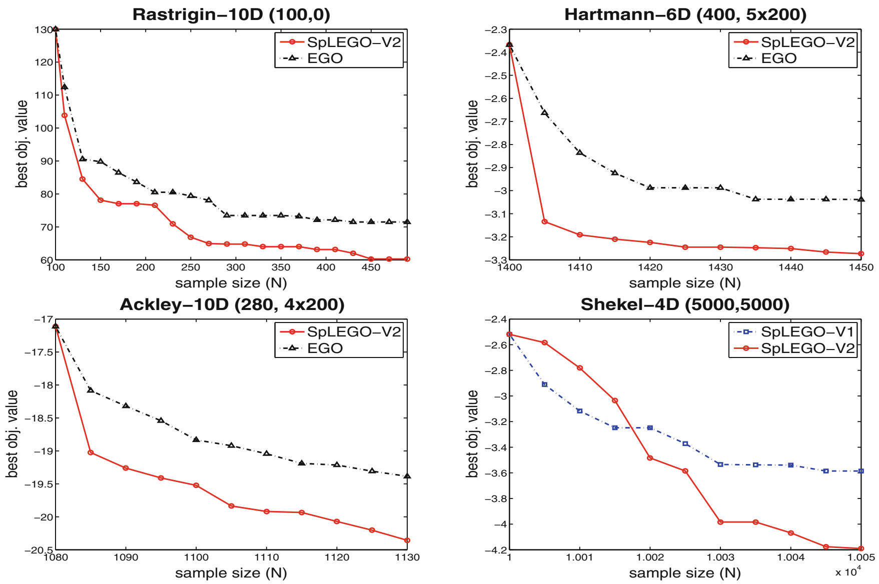
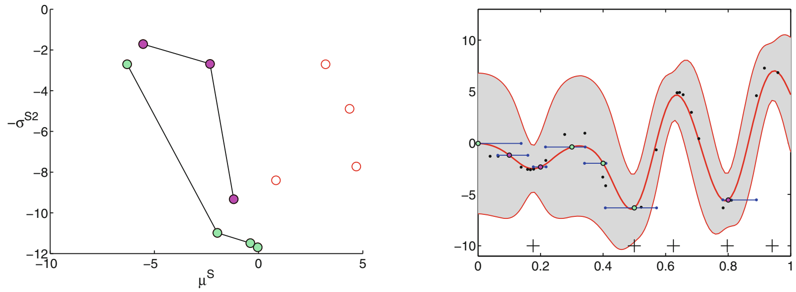


Fig. 4. Average best function value (10 trials) versus the number of sample size N. Initial sample size is $N_0 = N_1 + N_2$ where (N_1, N_2) is specified in each figure.

future research. One way to possibly improve the method is to incorporate also secondary non-dominated fronts in Step 3 when building local GP models and calculating local EIs. For example, in Fig. 5a, the first point on the secondary non-dominated front ($\mu^S \approx -5$, $-\sigma^{S2} \approx -2$) looks more promising than the last point on the Pareto front ($\mu^S \approx 0$, $-\sigma^{S2} \approx -12$). As expected, this point on the secondary front turns out to be the point at $x = 0.8$ in the decision space (Fig. 5b). In this example, we see that considering also the second front allows the algorithm to find a much wider spread of solutions.

In addition, Step 3c can be modified to allow SpLEGO to perform expensive function evaluations in a (synchronous or asynchronous) parallel way [7]. Instead of selecting only one point per iteration, several points from the pool of $\{w_i : i = 1, ..., r\}$ could be selected (in order of maximum local EI, from largest to smallest) and sent to the compute nodes. Once all the nodes have been taken, the next points wait in queue until the next node becomes available. Also, several candidate points may be considered for those regions with high potential, and possibly an arbitrage may be done between points from different subregions depending both on local EI and multipoint EI values [2].

One final remark about SPGP is that since $Md + |\Psi|$ hyperparameters need to be estimated, the standard SPGP method becomes no longer affordable for high dimensional data sets. Fortunately, [17] addresses this limitation by performing supervised dimensionality reduction in which the input space is projected to a low dimensional space. Problems of up to 10^2 dimensions and 10^4 number of training input points were considered in [17]. Consequently, this extension may



(a) A Pareto front for two objectives: SPGP mean and negative variance

(b) Pareto optimal point (green dots), second optimal (magenta dots), and its subregion R_i (blue line)

Fig. 5. Secondary non-dominated front and the corresponding subregions (Color figure online)

allow SpLEGO to be applied to solve global optimization problems where both the number of data points N and the input dimension d are large.

5 Conclusions

In this paper, SpLEGO, an extension of the EGO algorithm for handling a large number of starting points, was introduced and demonstrated on several test problems. SpLEGO is based on a multi-scale EI framework for global optimization that uses both sparse and local GP models. First, in the global scale, the space is partitioned using a Pareto-front approach with respect to the predictive mean and variance obtained from the sparse model. In the local scale, the algorithm zooms in specific regions around Pareto-optimal points, builds local GP models, and the next sample point is defined as the overall EI-optimal point among the maximizers of the several local EI criteria. The already obtained results demonstrate the effectiveness and robustness of our proposed method, yet there is still much room for improvement, particularly in developing adapted models and subregions for higher-dimensional problems, and also in handling the case of clustered points using some dedicated preliminary approach.

References

1. Ackley, D.H.: A Connectionist Machine for Genetic Hillclimbing. Kluwer, Dordrecht (1987)
2. Chevalier, C., Ginsbourger, D.: Fast computation of the multipoint expected improvement with applications in batch selection. In: Giuseppe, N., Panos, P. (eds.) Learning and Intelligent Optimization, pp. 59–69. Springer, Heidelberg (2014)
3. Csató, L., Opper, M.: Sparse on-line gaussian processes. Neural Comput. **14**(3), 641–668 (2002)

4. Dixon, L.C.W., Szegö, G.P.: The global optimization problem: an introduction. Towards Glob. Optim. **2**, 1–15 (1978)
5. Forrester, A.I.J., Keane, A.J.: Recent advances in surrogate-based optimization. Progr. Aerosp. Sci. **45**(1), 50–79 (2009)
6. Hansen, N., Finck, S., Ros, R., Auger, A., et al.: Real-parameter black-box optimization benchmarking 2009: noiseless functions definitions (2009)
7. Janusevskis, J., Le Riche, R., Ginsbourger, D., Girdziusas, R.: Expected improvements for the asynchronous parallel global optimization of expensive functions: potentials and challenges. In: Hamadi, Y., Schoenauer, M. (eds.) LION 2012. LNCS, vol. 7219, pp. 413–418. Springer, Heidelberg (2012)
8. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. J. Glob. Optim. **13**(4), 455–492 (1998)
9. Mockus, J.: Bayesian approach to global optimization. Springer, The Netherlands (1989)
10. Myers, R.H., Anderson-Cook, C.M.: Response Surface Methodology: Process and Product Optimization using Designed Experiments, vol. 705. Wiley, New York (2009)
11. Quiñonero-Candela, J., Rasmussen, C.E.: A unifying view of sparse approximate gaussian process regression. J. Mach. Learn. Res. **6**, 1939–1959 (2005)
12. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. The MIT Press, Cambridge (2006)
13. 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)
14. Roustant, O., Ginsbourger, D., Deville, Y.: DiceKriging, DiceOptim: two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization. J. Stat. Softw. **51**, 1–55 (2012)
15. Smola, A.J., Bartlett, P.: Sparse greedy gaussian process regression. In: Advances in Neural Information Processing Systems, vol. 13. Citeseer (2001)
16. Snelson, E., Ghahramani, Z.: Sparse gaussian processes using pseudo-inputs. In: Schölkopf, B., Weiss, Y., Platt, J. (eds.) Advances in Neural Information Processing Systems, vol. 18. MIT Press, Cambridge (2006)
17. Snelson, E., Ghahramani, Z.: Variable noise and dimensionality reduction for sparse gaussian processes. In: Proceedings of the 22nd International Conference on Uncertainty in Artificial Intelligence (2006)
18. Snoek, J., et al.: Scalable bayesian optimization using deep neural networks (2015). arXiv preprint arXiv:1502.05700
19. Sóbester, A., Leary, S.J., Keane, A.J.: On the design of optimization strategies based on global response surface approximation models. J. Glob. Optim. **33**(1), 31–59 (2005)
20. Stein, M.L.: Interpolation of Spatial Data: Some Theory for Kriging. Springer, New York (1999)
21. Titsias, M.K.: Variational learning of inducing variables in sparse gaussian processes. In: International Conference on Artificial Intelligence and Statistics, pp. 567–574 (2009)
22. Veenendaal, G.V.: Tree-GP: a scalable bayesian global numerical optimization algorithm. Master's thesis, Utrecht University, The Netherlands (2015)
23. Wang, G.G., Shan, S.: Review of metamodeling techniques in support of engineering design optimization. J. Mech. Des. **129**(4), 370–380 (2007)