Structural and Multidisciplinary Optimization (2021) 64:239–255
https://doi.org/10.1007/s00158-021-02882-7

RESEARCH PAPER

Sparsifying to optimize over multiple information sources: an augmented Gaussian process based algorithm

Antonio Candelieri[1] · **Francesco Archetti**[2]

Received: 23 June 2020 / Revised: 18 September 2020 / Accepted: 11 February 2021 / Published online: 5 April 2021

Abstract
Optimizing a black-box, expensive, and multi-extremal function, given multiple approximations, is a challenging task known as multi-information source optimization (MISO), where each source has a different cost and the level of approximation (aka *fidelity*) of each source can change over the search space. While most of the current approaches *fuse* the Gaussian processes (GPs) modelling each source, we propose to use GP *sparsification* to select only "reliable" function evaluations performed over all the sources. These selected evaluations are used to create an augmented Gaussian process (AGP), whose name is implied by the fact that the evaluations on the most expensive source are *augmented* with the reliable evaluations over less expensive sources. A new acquisition function, based on confidence bound, is also proposed, including both cost of the next source to query and the location-dependent approximation of that source. This approximation is estimated through a *model discrepancy* measure and the prediction uncertainty of the GPs. MISO-AGP and the MISO-fused GP counterpart are compared on two test problems and hyperparameter optimization of a machine learning classifier on a large dataset.

Keywords Multi-information source optimization · Gaussian process · Sparsification · Machine learning

1 Introduction

1.1 Overview

This paper focuses on the situation arising when a black-box, multi-extremal, and expensive function can be optimized by querying multiple information sources which provide less expensive approximations of the original function. The final goal is to optimize the original function while keeping low the overall cumulated query cost. This setting is known as multi-information source optimization (MISO).

When the different sources come with an explicit information about their level of approximation, usually named *fidelity*, MISO specializes in *multi-fidelity* optimization, first introduced in Kennedy and O'Hagan (2000). Knowledge about fidelities can be exploited to sort hierarchically the sources in order to implement efficient and effective multi-fidelity optimization methods (Peherstorfer et al. 2017; Sen et al. 2018; Marques et al. 2018; Chaudhuri et al. 2019; Kandasamy et al. 2019). However, as already reported in March and Willcox (2012), a number of drawbacks can arise with respect to sources hierarchically organized: once one has queried a fidelity source at a location x, no further knowledge can be obtained querying any other source of lower fidelity, at any location. Moreover, hierarchical organization requires the assumption that information sources are unbiased, admitting only *aleatoric* error that must be independent across sources.

The mentioned drawbacks were addressed first in Lam et al. (2015) who proposed an approach to generate a single model integrating the different information sources with fidelities changing over the search space. Thus, sources are not necessarily unbiased and independent and allow for *epistemic* error. More recently, Poloczek et al. (2017) introduced a general notion of *model discrepancy* to quantify the difference between each source and the function to optimize, depending on the location. In MISO, model discrepancy is different from the aleatoric uncertainty, and it

Responsible Editor: Joaquim R. R. A. Martins

✉ Antonio Candelieri
antonio.candelieri@unimib.it

[1] Department of Economics, Management and Statistics, University of Milano-Bicocca, Milan, Italy

[2] Department of Computer Science, Systems and Communication, University of Milano-Bicocca, Milan, Italy

is considered a residual process to mitigate structural bias (Liu et al. 2019).

The reference problem for MISO is:

$$x^* = \arg\min_{x \in \Omega \subseteq \Re^d} f(x) \quad (1)$$

with $f(x)$ black-box, multi-extremal, and expensive, and Ω the search space. Minimization is considered without loss of generality (i.e., $\max f(x) = \min -f(x)$).

Problem (1) is the reference problem for global optimization: what makes MISO a different problem is the availability of *sources* approximating $f(x)$, with a different cost for querying each of them. All the sources are also black-box and potentially multi-extremal.

Let $\{f_1(x), ..., f_s(x), ..., f_S(x)\}$ denote the S different sources available, where $s = 1$ identifies the most expensive source: in the case that also $f(x)$ can be queried we have $f_1(x) = f(x)$. In the rest of this paper, we make this assumption without any loss of generality.

Let c_s denote the cost for querying the source $f_s(x)$, with $c_s > 0 \; \forall s = 1, ..., S$. One can always sort sources so that $c_s > c_{s+1}$. Since fidelity changes over the search space, this cost-based ranking does not imply a fidelity-based ranking or hierarchy of the sources.

Querying the source s at a certain location $x \in \Omega$ leads to the observation $y_s = f_s(x)$, or $y_s = f_s(x) + \varepsilon_s$ in the case of a noisy setting, with $\varepsilon_s \sim \mathcal{N}(0, \lambda_s^2)$.

Bayesian optimization (BO) (Shahriari et al. 2015; Frazier 2018; Archetti and Candelieri 2019) is a mathematically principled and sample efficient global optimization algorithm. A lot of research has been made to extend the advantages offered by BO to MISO, especially by using Gaussian process (GP) modelling to approximate the sources and then choose, sequentially, the next *source-location* pair to query. GP modelling (Williams and Rasmussen 2006) is one of the widely adopted kernel-based learning algorithms for both regression and classification tasks. A GP is a probabilistic model, able to provide a prediction along with an estimate of its uncertainty. Unfortunately, GP training is not scalable on large datasets because its computational complexity is $\boldsymbol{O}(n^3)$, with n the number of examples (aka instances) into the dataset. Computational complexity of GP is a quite relevant issue in "learning" tasks, representing a relevant gap to its applicability on large datasets. Thus, *GP approximation* is still a relevant topic in the machine learning (ML) and statistical learning community, as also recently addressed in Schreiter et al. (2016).

GP approximation—specifically *GP sparsification*—is considered in this study not for reducing computational costs of GP fitting in MISO, but as the core of our algorithm in order to reduce the discrepancy between $f(x)$ and the single model enabling a uniform Bayesian treatment of all the information sources. More specifically, the set to be sparsified is that consisting of all the function evaluations performed over all the information sources. GP sparsification methods are described in Section 2.2; here, it is important to anticipate that they are aimed at restricting the GP model to a small set of *inducing locations* which should be large enough to cover the search space in order to avoid *variance starvation* (Wang et al. 2018), and also be as small as possible for efficiency.

1.2 Related works

MISO and multi-fidelity optimization have been gaining increasing attention in the last years in many real-life problems, especially in aerodynamics (Feldstein et al. 2019; Marques et al. 2020), where simulation models are considered less expensive information sources of the actual aerodynamic system to optimize. Additional information sources can be physical prototypes, with costs depending on the experimental setting.

Another domain whose importance for MISO and multi-fidelity optimization has been growing is ML, especially automated machine learning (AutoML) (Hutter et al. 2019) and neural architecture search (NAS) (Elsken et al. 2019). Indeed, searching for the best ML algorithm and the optimal configuration of its hyperparameters might take hours or even days. The seminal paper adopting MISO in ML is Swersky et al. (2013), which proposes a method to use small datasets to quickly optimize the hyperparameters of ML algorithms on a large dataset. Results proved that it is possible to transfer the knowledge gained from previous optimizations to new tasks in order to speed up k-fold cross-validation. Successively, Klein et al. (2017) proposed FABOLAS (FAst Bayesian Optimization on LArge dataSets), an approach for hyperparameter optimization on large datasets that selects hyperparameter values and a dataset size, iteratively, in order to identify optimal hyperparameter values for the entire large dataset.

From the global optimization perspective, Lam et al. (2015) was the first paper addressing location-dependent fidelities of the sources, removing the assumption about hierarchical relations across them. The approach uses a separate GP for each information source and then *fuse* their predictions—and associated uncertainties—through the method proposed by Winkler (1981), which came to represent the standard practice for the fusion of normally distributed data. The detailed process of estimating the correlation between the errors of two models, at the basis of the fusing procedure, is given in Thomison and Allaire (1949).

The approach proposed in Poloczek et al. (2017) uses a GP to capture the model discrepancy of each information source with respect to $f(x)$, while a single statistical model is used to perform BO jointly on the search space and

the information sources. A kernel able to deal with both *location* and *source* is used to exploit correlations across different information sources. This allows reducing the uncertainty on all the information sources whenever a new function evaluation is performed, even if it comes from a less accurate source.

Analogous to mentioned approaches, also Ghoreishi and Allaire (2019) use a GP for each information source and fuse them into a single statistical model. More precisely all the GPs are *fused* through the method by Winkler (1981), as previously proposed also by Lam et al. (2015). The main contribution is the adoption of a two-step look ahead acquisition function for the selection of the next *source-location* pair to query. To our knowledge Ghoreishi and Allaire (2019) is the most recent and complete MISO approach, since it also considers black-box constraints. The main drawback in *fusing* GPs is that the computation of correlations requires using a further set of N_f points, randomly selected, which determines both the computational complexity and the smoothness of the resulting fused GP. For instance, in Ghoreishi and Allaire (2019), N_f locations are used, even if the authors do not provide any information on how they have chosen this value.

1.3 Our contribution

The main contributions of this paper can be summarized as follows:

- A new mechanism for generating a single model on the information sources, based on GP sparsification instead of *fusion*. A low complexity criterion (i.e., with complexity $O(1)$) is introduced to decide whether a function evaluation performed on a cheap source can be selected to "augment" the set of function evaluations on $f_1(x)$. The GP fitted on the augmented dataset is called *Augmented GP* (AGP): this entails a lower computational complexity than fusing GP (Lam et al. 2015; Ghoreishi and Allaire 2019).
- A new acquisition function to select the next *source-location* pair, by mixing together: the GP Confidence Bound, the cost of the source and the (location dependent) model discrepancy between the source specific GP and the Augmented GP.
- Avoiding variance starvation, premature convergence to local optima, as well as ill-conditioning in the GP training, by replacing, if needed, our acquisition function with a variance maximization step on the most expensive source (as discussed in Section 3.3).
- Computational experiments to confirm the actual performance of the MISO-AGP method on benchmark functions and the hyperparameter optimization of a SVM classifier on a large dataset.

The rest of the paper is organized as follows: Section 2 is devoted to the methodological background about GP regression, GP sparsification and Bayesian optimization. Section 3 is devoted to the structure of the proposed MISO-AGP algorithm. Section 4 is devoted to the experimental setting and Section 5 to the computational results.

2 Background

2.1 Gaussian process regression

A GP is a random function $f : \Omega \rightarrow \Re$ whose outputs are drawn from a multivariate normal distribution, that is $f(x) \sim \mathcal{N}(\mu(x), \sigma^2(x))$. When n function evaluations of $f(x)$ have been already performed, the values of $f(x)$ can be conditioned on them through the GP posterior. Let $\mathbf{X_{1:n}} = \{x_1, ..., x_n\}$ and $\mathbf{y} = \{y, ..., y_n\}$ denote, respectively, the n points evaluated so far and the associated observed values, then the mean and variance of the multivariate normal distribution can be computed as follows:

$$\begin{aligned} \mu(x) &= \mathrm{k}(x, \mathbf{X_{1:n}}) \left[\mathbf{K} + \lambda^2 \mathbf{I}\right]^{-1} \mathbf{y} \\ \sigma^2(x) &= k(x, x) - \mathrm{k}(x, \mathbf{X_{1:n}}) \left[\mathbf{K} + \lambda^2 \mathbf{I}\right]^{-1} \mathrm{k}(\mathbf{X_{1:n}}, x) \end{aligned} \tag{2}$$

where λ^2 is the variance of—a zero-mean Gaussian—noise in the case of noisy observations, $\mathbf{K} \in \Re^{n \times n}$, such that $\mathbf{K_{ij}} = k(x_i, x_j)$, with k a *kernel* function modelling the covariance in the GP. Finally, $\mathrm{k}(x, \mathbf{X_{1:n}})$ is a vector whose i-th component is given by $k(x, x_i)$ (for completeness, $\mathrm{k}(\mathbf{X_{1:n}}, x) = \mathrm{k}(x, \mathbf{X_{1:n}})^\top$).

Different types of kernels are available, such as squared exponential (aka Gaussian), and Matérn and exponential (aka Laplacian). Each kernel has its own hyperparameters which are typically fitted on data through maximum log-likelihood estimation (MLE) or maximum a posteriori estimation (MAP).

While the kernel type establishes a prior on the structural characteristics of the approximation (e.g., the Laplacian kernel leads to $f(x)$ that are not continuously differentiable), the values of the kernel's hyperparameters allow modulating the amplitude and the smoothness of $f(x)$, conditioned to the observations but anyway "constrained" by the kernel type chosen.

In this paper, we use the squared exponential (SE) kernel, $k_{SE}(x, x') = \sigma^2_{SE} e^{\frac{\|x - x'\|^2}{2\ell^2}}$ and MLE to fit its hyperparameters σ^2_{SE} and ℓ, namely kernel's output variance and lengthscale, respectively.

The formulation (2) is known as *function-space view* and it is easy to understand that the most computationally expensive operation is the matrix inversion, with computational complexity $\boldsymbol{O}(n^3)$.

Another possible formulation for GP modelling is the so-called *weight-space view*, which approximates $g(x)$ through an explicit set of basis functions. According to the *kernel trick* (Scholkopf and Smola 2001), the kernel function $k(\cdot,\cdot)$ can be considered the inner product in a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ equipped with a feature map function $\varphi : \Omega \rightarrow \mathcal{H}$ such that $k(x,x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}}$. If $\mathcal{H}$ is separable, then the kernel function can be approximated through an explicit feature map function $\phi : \Omega \rightarrow \Re^q$, such that:

$$k(x,x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}} \approx \phi(x)^\top \phi(x') \tag{3}$$

For stationary kernel functions, Bochner's theorem provides a suitable q-dimensional feature map based on a set of random Fourier features (Rahimi and Recht 2008). More precisely, $\phi_i(x) = \sqrt{\frac{2}{q}} \cos(\boldsymbol{\theta}_i x + \tau_i)$, with $\boldsymbol{\theta}_i$ sampled assuming the (scaled) kernel's spectral density as probability distribution and $\tau_i \sim \mathbf{U}(0, 2\pi)$. Then, $f(x)$ is approximated as follows:

$$f(x) = \sum_{i=1}^{q} w_i \phi_i(x) \tag{4}$$

where weights w_i are sampled from a posterior distribution conditioned to the observations collected so far, that is $w_i \sim \mathcal{N}(\mu_w, \sigma_w^2)$ with:

$$\begin{aligned} \mu_w &= \left(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \lambda^2 \mathbf{I}\right)^{-1} \boldsymbol{\Phi}^\top y \\ \sigma_w &= \left(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \lambda^2 \mathbf{I}\right)^{-1} \lambda^2 \end{aligned} \tag{5}$$

and where $\boldsymbol{\Phi}$ is an $n \times q$ matrix whose i-th row is given by $\phi(x_i) = [\phi_1(x_i), ..., \phi_q(x_i)]$ with $x_i \in \mathbf{X_{1:n}}$.

Typically, $q \ll n$, so the computational complexity for fitting the GP—according to (4)—is reduced to $\boldsymbol{O}(q^3)$ due to the inversion of the matrix $\left(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \lambda^2 \mathbf{I}\right)$ in (5). Thus, the weight space view leads naturally to a GP approximation whose quality is controlled through q: the greater the value of q the better the approximation (but the higher the computational cost). This *kernel approximation* method is—together with the Nystrøm method—a *covariance matrix approximation* (Schreiter et al. 2016). GP approximation methods have been also proposed for the function space view and are usually known as *sparse likelihood approximation* techniques (Schreiter et al. 2016) or, more commonly, *GP sparsification* methods. Recently, the combination of the two "views" has been proposed to perform efficient function sampling from a GP (Wilson et al. 2020).

In this paper, the focus is on GP sparsification which is at the basis of the proposed MISO-AGP approach. A brief overview of GP sparsification methods is presented in the following sub-section.

2.2 Gaussian process sparsification

GP sparsification methods are aimed at restricting the GP model to a small set of *inducing locations*. This set should be large enough to cover the search space and to avoid variance starvation (Wang et al. 2018) but, on the other way around, as small as possible in order to make GP modelling scalable on large datasets. Many different methods have been proposed for selecting the set of inducing locations (Wahba 1990; Csató and Opper 2001; Smola and Bartlett 2001; Csató and Opper 2002; Seeger et al. 2003; Seeger 2008; Schreiter et al. 2016). In particular, Smola and Bartlett (2001) is one of the best approaches in terms of model accuracy but it is heavily inefficient. It was successively improved in Keerthi and Chu (2006) in terms of computational complexity but significantly increases the memory requirements, especially on large and high-dimensional datasets.

Two early papers proposed an original approach on the issue of sparsification by elaborating criteria for *deletion* (Csató and Opper 2001) and *insertion* (Csató and Opper 2002) of observations from and into the *basis vector set* (i.e., inducing locations), for which the exact update is performed.

Then, Seeger et al. (2003) presented a method which randomly selects the set of inducing points, called *support patterns*, based on information gain and Kullback–Leibler divergence, leading to a sufficiently stable approximation of the marginal log-likelihood of the training data.

Successively, Seeger (2008) argues that, beyond the analysis of large datasets, the value of Bayesian modelling is larger in higher level tasks such as making optimally cost-efficient decisions or experimental designs where data is sampled in a sequential and actively controlled manner. Seeger's main argument is that GPs do not encode sparsity and reports substantial benefits using Laplace prior, such as to offset the absence of analytically tractable formulae for inference. A key component in the choice of the selection of inducing points is the expected information gain, for whose computation Seeger suggests a computationally effective approximation. Following the selection of the support patterns, Gaussian properties of the approximation are retrieved by adopting a Gaussian posterior (i.e., a Gaussian kernel). This closely resembles BO which also is an *active learning* approach because it selects the new points according to their informative value.

2.3 Bayesian optimziation

BO (Shahriari et al. 2015; Frazier 2018; Archetti and Candelieri 2019) is a sample efficient strategy for solving the problem (1) under a limited number of function evaluations available. BO consists of two components:

(i) a *probabilistic surrogate model*, fitted on the function evaluations performed so far, approximating the objective function $f(x)$, and (ii) an *acquisition function* (aka *infill criterion* or *utility function*) which drives the choice of the next location x to evaluate while dealing with the *exploitation-exploration* dilemma.

The probabilistic surrogate model provides an estimate of $f(x)$ along with a measure of uncertainty about such an estimate. The estimate is usually given by the prediction mean of the model, $\mu(x)$, while prediction uncertainty is given by the standard deviation, $\sigma(x)$.

GP is the common choice for the probabilistic surrogate model, especially when the search space is continuous. Random Forest (Goel et al. 2017) is the most common alternative, especially when the search space is spanned by discrete, mixed as well as conditional variables, which is a common situation in AutoML.

Several acquisition functions have been proposed, offering different strategies for balancing exploitation and exploration in selecting the next point to evaluate (an overview is reported in Shahriari et al. (2015), Frazier (2018), and Archetti and Candelieri (2019)).

However, BO—as is—is not well suited for MISO because it does not exploit the availability of different sources in order to keep low the overall querying cost over the sequential optimization process.

3 MISO via augmented GP

3.1 Fitting an augmented Gaussian process

Let us denote with $D_s = \{(x^{(i)}, y^{(i)})\}_{i=1,\ldots,n_s}$ the set of function evaluations performed so far on the source s. The final aim in MISO is to solve (1) using the cheaper approximations to reduce the overall cost over the entire optimization process.

The basic idea of the proposed MISO-AGP algorithm (MISO via augmented GP) is to use GP sparsification for selecting a subset of the function evaluations—among those performed so far over all the different sources—as inducing locations to generate the AGP approximating $f(x)$. The GP sparsification proposed is an *insertion* method: the set of inducing locations is initialized with the function evaluations on $f_1(x)$ (i.e., the most expensive information source) and is incremented by including evaluations on other sources depending on both a model discrepancy measure and GP's predictive uncertainty. It is important to remark that the proposed MISO-AGP algorithm works on the assumption that $f(x) = f_1(x)$.

Denote with $\eta(\mathcal{G}, \mathcal{G}', x)$ the model discrepancy between two GPs, $\mathcal{G}$ and $\mathcal{G}'$, at a certain location x. We compute $\eta(\mathcal{G}, \mathcal{G}', x)$ simply as the absolute difference between the two means of the GPs:

$$\eta(\mathcal{G}, \mathcal{G}', x) = |\mu(x) - \mu'(x)| \tag{6}$$

This simplified measure does not require to compute the discrepancy as a further model, such as in Poloczek et al. (2017) where it is modelled as a separate GP.

Then, the set of inducing locations, denoted with $\widehat{D}$, is computed as $\widehat{D} \leftarrow D_1 \cup \bar{D}$, where D_1 is the dataset related to all the function evaluations performed so far on $f_1(x)$ and $\bar{D}$ is defined as follows:

$$\bar{D} = \{(\bar{x}, \bar{y}) : \exists \bar{s} : (\bar{x}, \bar{y}) \in D_{\bar{s}} \wedge \eta(\mathcal{G}_1, \mathcal{G}_{\bar{s}}, \bar{x}) < m\sigma_1(\bar{x})\} \tag{7}$$

where m is the first of two technical parameters of the MISO-AGP algorithm. Basically, a function evaluation performed at location $\bar{x}$ on a cheap source $\bar{s} \neq 1$ is *inserted* into the set of inducing locations only if the discrepancy between $\mathcal{G}_1$ and $\mathcal{G}_{\bar{s}}$ at $\bar{x}$ is lower than m times the standard deviation of $\mathcal{G}_1$ at $\bar{x}$. Here, we are considering that $\sigma^2_{\bar{s}}(\bar{x})$ is close to 0—or anyway limited by the noise variance $\lambda^2_{\bar{s}}$ in the case of a noisy setting.

In Fig. 1, we report an example about how (6) and (7) work. On the top, three GPs are fitted according to function evaluations performed on the associated information sources. On the bottom: the two solid lines are

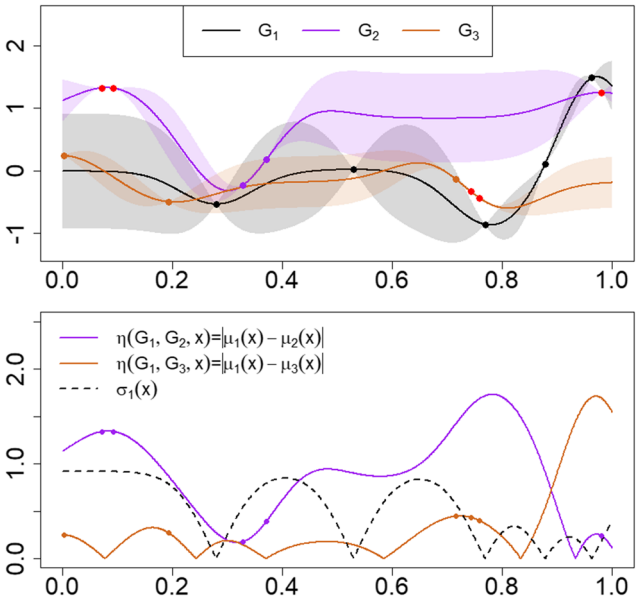

Fig. 1 Selection of the inducing locations for the AGP. Top: three GPs fitted on the function evaluations performed on the associated sources. Bottom: solid lines are discrepancies between the GP on the most expensive source (i.e., $\mathcal{G}_1$) and the GPs related to the other two sources (i.e., purple for $\eta(\mathcal{G}_1, \mathcal{G}_2, x)$ and light brown for $\eta(\mathcal{G}_1, \mathcal{G}_3, x)$). The dashed line is the standard deviation of $\mathcal{G}_1$ (i.e., $\sigma_1(x)$): all the function evaluations on cheap sources having a discrepancy greater than $\sigma_1(x)$ are not selected as inducing locations for the AGP (i.e., red dots in the top)

the discrepancies—computed according to (6)—between the GP of the most expensive source, namely $\mathcal{G}_1$ in the figure, and the GPs of the other two sources. All the function evaluations on cheap sources having a discrepancy lower than the $\mathcal{G}_1$'s standard deviation (i.e., $m = 1$) are selected, according to (7), to "augment" the function evaluations on $f_1(x)$, leading to the set of the AGP's inducing locations $\widehat{D}$, while all the others are discarded (i.e., red dots on the top of the figure). With respect to computational complexity, the cost of our selection strategy is $O(1)$, exactly as the most efficient presented in Seeger et al. (2003) and Schreiter et al. (2016).

The suggested value is $m = 1$ because around 68% of the possible values for $f_1(x)$ are estimated to be within $\mu_1(x) \pm \sigma_1(x)$ (i.e., around 95% in $\mu_1(x) \pm 2\sigma_1(x)$ and around 99% in $\mu_1(x) \pm 3\sigma_1(x)$). Indeed, increasing m leads to including more evaluations performed on the cheap sources even if they have a high discrepancy and, consequently, might degrade the approximation offered by the AGP. In Fig. 2, we show an example about the role of m in fitting the AGP. From left to right, the increasing value of m (i.e., $m = 1, .., 3$) implies the selection of more evaluations performed on cheap sources, significantly changing the resulting AGP. Moreover, the most optimistic estimation the minimum of $f_1(x)$ (i.e., $\widehat{\mu}(x) - \widehat{\sigma}(x)$), drastically changes, leading to a prediction of the global minimizer of $f_1(x)$ far away from its actual location. Indeed, in Fig. 2, for $m = 1$, the estimated location of the most optimistic minimum is quite close to the actual global minimizer, that is $x = 0.7572488$ (as reported in Section 4), while for $m = 2$ and $m = 3$, the location is around $x = 0$, very far from the actual location.

Finally, the AGP $\widehat{\mathcal{G}}$ is fitted on the inducing locations $\widehat{D}$, leading to $\widehat{\mu}(x)$ and $\widehat{\sigma}^2(x)$ according to equations in (2). The resulting AGP is very different from models proposed in previous studies which use all the function evaluations on all the sources to generate a unified discrepancy model (Poloczek et al. 2017) or to *fuse* the GPs into a single model over the different sources (Lam et al. 2015; Ghoreishi and Allaire 2019).

In Fig. 3, we compare the AGP and the *fused* GP, both fitted starting from the same set of function evaluations performed on three different sources. The equations of the three sources are from Ghoreishi and Allaire (2019); we have changed their sign with respect to the original ones because in our paper we consider a minimization setting:

$$\begin{aligned} f_1(x) &= (1.4 - 3x)sin(18x) \\ f_2(x) &= (1.6 - 3x)sin(18x) \\ f_3(x) &= (1.8 - 3x)sin(18x + 0.1) \end{aligned} \tag{8}$$

Our insertion mechanism, based on (6) and (7), selects only a subset of the overall function evaluations as inducing points (blue dots). On the contrary, fused GP uses all of them: our AGP is more accurate in approximating $f_1(x)$ around the global minimizer, and it is less affected by variance starvation and less computationally expensive.

3.2 Selecting the next *source-point* to query

In MISO, the next selection consists of a *source-location* pair, (s', x'). Thus, traditional acquisition functions used in BO are not well-suited. In MISO-AGP, the proposed acquisition function considers, for every source $s = 1, ...S$ and every location $x \in \Omega$, the most *optimistic* improvement with respect to the best value observed so far among the current inducing locations of the AGP, then penalizes this improvement depending on source's cost, c_s, and the discrepancy—computed as in (6)—between the GP associated to the source, $\mathcal{G}_s$, and the AGP, $\widehat{\mathcal{G}}$:

$$\alpha_s(x, \widehat{y}^+) = \frac{\widehat{y}^+ - \left[\widehat{\mu}(x) - \sqrt{\beta^t}\widehat{\sigma}(x)\right]}{c_s\left(1 + \eta(\widehat{\mathcal{G}}, \mathcal{G}_s, x)\right)} \tag{9}$$

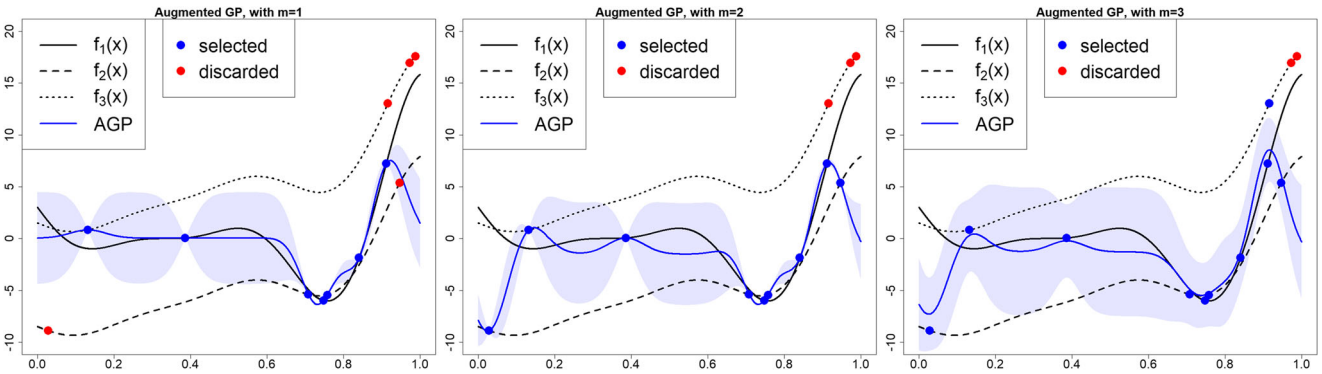

Fig. 2 Role of m in the AGP fitting process: an example on three sources with four function evaluations on each source. The sources are related to the first experiment presented in Section 4. When increasing from $m = 1$ (left) to $m = 2$ (middle) and up to $m = 3$ (right), more evaluations on cheap sources (i.e., $f_2(x)$ and $f_3(x)$) are selected, along with those on $f_1(x)$, as inducing locations of the AGP. Consequently, the approximation offered by AGP might result poorer, especially in terms of the most optimistic estimation of the $f_1(x)$ optimum, that is $\widehat{\mu}(x) - \widehat{\sigma}(x)$, aka lower confidence bound

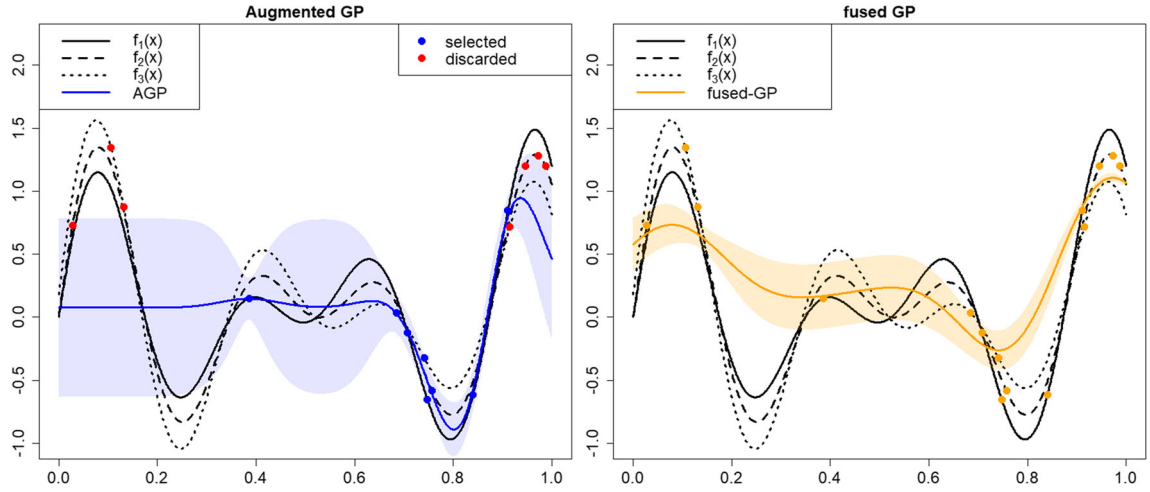

Fig. 3 A comparison between the proposed AGP (left) and the fused GP (right) over three sources: the evaluations in red are explicitly discarded to fit the AGP while, on the contrary, all the evaluations are used to fit the fused GP, including those outside the yellow shaded area delimited by mean ± standard deviation

We talk about the most *optimistic* improvement because $\widehat{y}^+$ is the best value observed so far (usually named *best seen*) and the quantity $\left[\widehat{\mu}(x) - \sqrt{\beta^t}\widehat{\sigma}(x)\right]$ is the well-known *lower confidence bound* of $\widehat{\mathcal{G}}$, with β^t the parameter regulating the exploration-exploitation trade-off and t the number of function evaluations used to fit the GP (a scheduling for β^t and a convergence proof are given in Srinivas et al. (2012)). In the denominator, we use $1 + \eta(\widehat{\mathcal{G}}, \mathcal{G}_s, x)$ to avoid division-by-zero; $c_s > 0 \; \forall s = 1, ..., S$ by definition.

It is easy to demonstrate that maximizing (9) is equivalent to minimizing the lower confidence bound of the AGP $\widehat{\mathcal{G}}$, divided by $c_s(1 + \eta(\widehat{\mathcal{G}}, \mathcal{G}_s, x))$. We have decided to use the notation in (9) for the sake of homogeneity with the procedure described in the following Section 3.3 and aimed at correcting, if needed, the selected *source-location* pair. We have decided to base our acquisition function on GP confidence bound because its computational cost is lower compared to other look-ahead acquisition functions proposed in other papers.

It is important to note that, due to the GP sparsification performed at every iteration, the best seen $\widehat{y}^+$ in MISO-AGP has a different behavior from BO and other MISO approaches which do not use GP sparsification (Lam et al. 2015; Poloczek et al. 2017; Ghoreishi and Allaire 2019). At every iteration of MISO-AGP, the current set of inducing locations is identified by using (6) and (7), and $\widehat{y}^+$ is computed as the best value observed among the current inducing locations. As a consequence, the curve obtained by plotting $\widehat{y}^+$ with respect to the function evaluations is not strictly monotone because the $\widehat{y}^+$ obtained at a certain iteration could be not selected as inducing location at successive ones. To distinguish $\widehat{y}^+$ in MISO-AGP from the common best seen—which is usually denoted with y^+—we named it *augmented best seen*.

For completeness, we report here the mathematical formulations of the best seen y^+—used in BO and MISO-fused GP—and the *augmented best seen* $\widehat{y}^+$. All the formulations are intended at a generic iteration.

BO best seen BO (single source)

$$y^+ = \min_{i=1:n} \left\{ y^{(i)} \right\}, \text{ where } y^{(i)} : \exists \left(x^{(i)}, y^{(i)} \right) \in D_1 \tag{10}$$

where n is the number of function evaluations performed so far

MISO-fused GP best seen

$$y^+ = \min_{\substack{s=1:S \\ i=1:n_s}} \left\{ y^{(i)} \right\}, \text{ where } y^{(i)} : \exists \left(x^{(i)}, y^{(i)} \right) \in D_s \tag{11}$$

where n_s is the number of function evaluations performed so far on source s

MISO-AGP *augmented* best seen

$$\widehat{y}^+ = \min_{i=1:p} \left\{ y^{(i)} \right\}, \text{ where} y^{(i)} : \exists \left(x^{(i)}, y^{(i)} \right) \in \widehat{D} \tag{12}$$

where p is the number of inducing locations in $\widehat{D}$ at the current iteration.

We remark that the sequences related to (10) and (11), over function evaluations, are monotonically decreasing, while this is not true for the sequences related to (12)

Finally, the next pair (s', x') to evaluate is obtained by solving the following auxiliary problem:

$$(s', x') \leftarrow \arg\max_{\substack{x \in \Omega \\ s=1,...,S}} \alpha_s(x, \widehat{y}^+) \tag{13}$$

with $\alpha_s(x, \widehat{y^+})$ defined as in (9). A representation of the proposed acquisition function is depicted in Fig. 4, applied to the AGP and fused GP of previous Fig. 3. In both the two cases, the cheapest source will be selected as s', but the location x' chosen using the AGP will be closer to the global optimizer (i.e., around 0.8 and 0.75 for AGP and fused GP, respectively).

However, solving the auxiliary problem (13) could lead to select an x' very close to some location already evaluated on the source s'. This could occur especially when the optimization process is converging towards a local optimum, making instable the inversion of the matrix $K + \lambda^2 I$, especially in the noise-free setting. To overcome this undesired situation, we propose the correction reported in the following section.

3.3 Correcting the *source-point* pair

The ill-conditioning of the matrix $K + \lambda^2 I$ is a well-known issue in GP modelling. Some basic workarounds consist in adding some noise to the observed value or to simply select a different location in the case that ill-conditioning occurs. Our correction is aligned with the second option: basically,

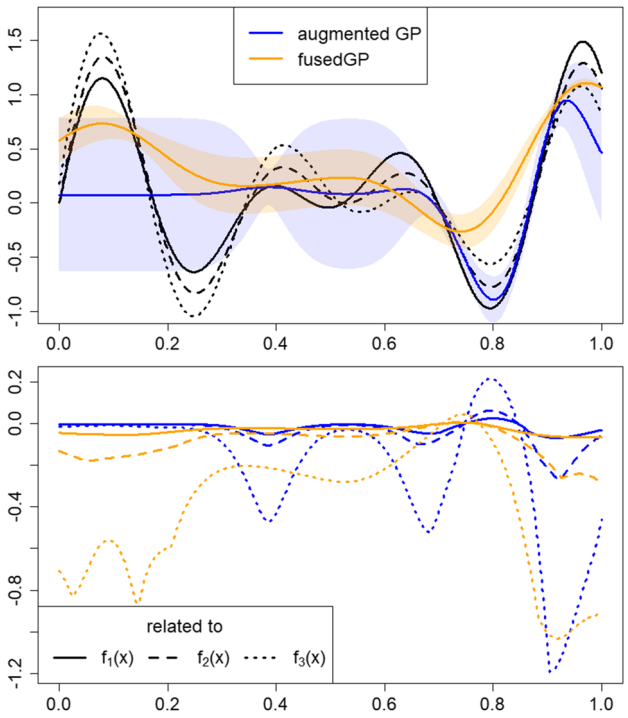

Fig. 4 An illustration of the acquisition function (9), with respect to AGP and fused GP, respectively. Top: sources, AGP, and fusedGP, as reported in previous Fig. 3. Bottom: the value of the acquisition function. Line type (solid, dashed, and dotted) is used to differentiate among sources, while color is used to differentiate between AGP and fused GP. According to the situation in the figure, both AGP and fused GP will select $s' \leftarrow 1$, but two different locations (i.e., around 0.8 and 0.75, for AGP and fused GP, respectively)

we discard x' if it is close to a previous observation on s' more than a given threshold $\delta > 0$—that is the second technical parameters of the MISO-AGP algorithm. In our approach, we propose to choose an alternative x' by "investing" on *function learning*, meaning that x' should be a location improving the approximation offered by the AGP at the next iteration. Under this consideration, we propose to set $s' = 1$ and then choose the location x' associated to the highest predictive uncertainty of $\mathcal{G}_1$.

Correction *If* $\exists(\tilde{x}, \tilde{y}) \in D_{s'} : ||\tilde{x} - x'|| < \delta$ *then set* $s' \leftarrow 1$ *and choose* x' *as follows:*

$$x' \leftarrow \arg\max_{x \in \Omega} \sigma_1(x) \tag{14}$$

As stated in Srinivas et al. (2012), solving (14) allows to globally explore the search space and, consequently, is a good strategy for *function learning* (aka *function approximation*), whose goal is to efficiently sample $f(x)$ at different locations in order to obtain an accurate approximation $\hat{f}(x)$ with a limited number of function evaluations. A Smart Sampling and Incremental Function Learning (SSIFL) algorithm has been proposed in Pedergnana et al. (2016), where the method is also compared with other typical strategies, also not sequential, from the experiment design field.

Basically, while BO can be considered *active learning* aimed at finding, as fast as possible, the optimizer of a black-box function, function learning can be considered active learning aimed at obtaining, as fast as possible, an accurate approximation of a black-box function, over the whole search space Ω. This is the important difference, making (14) a good choice to *explore* $f(x)$ globally but not well-suited for BO, which instead requires to focus sampling eventually near the global optimum.

These considerations led to the definition of the GP confidence bound acquisition function in Srinivas et al. (2012). Indeed, (14) is guaranteed to provide a near-optimal solution for the global optimization of a black-box function. More precisely, the informativeness of a set of sampling points $\mathbf{X_{1:n}} \in \Omega$ is measured by the *information gain*, which is the mutual information between $f(x)$ and the observations $\mathbf{y} = f(\mathbf{X_{1:n}}) + \varepsilon$, with $\varepsilon \sim \mathcal{N}(0, \lambda^2 \mathbf{I})$.

In the GP setting, we have $\mathrm{I}(\mathbf{y}, f(\mathbf{X_{1:n}})) = \frac{1}{2}\log|\mathbf{I} + \lambda^2 \mathbf{K_X}|$, where $\mathbf{K_X} = [k(x_i, x_j)]_{x_i, x_j \in \mathbf{X_{1:n}}}$. Finding the information gain maximizer $\mathbf{X_{1:n}}$, with $n \leq N$, is NP-hard. An approximate solution is given by an efficient procedure consisting in sequentially selecting the next location as $x' \leftarrow \arg\max_{x \in \Omega} F(X \cup \{x\})$, where $F(\mathbf{X_{1:n}}) = \mathrm{I}(\mathbf{y}, f)$. Being a sequential strategy, after each selection, we update: $\mathbf{X_{1:n+1}} \leftarrow \mathbf{X_{1:n}} \cup \{x'\}$, $\mathbf{y} \leftarrow \mathbf{y} \cup \{f(x') + \varepsilon\}$ and $n \leftarrow n + 1$.

This is equivalent to select $x' \leftarrow \arg\max_{x \in \Omega} \sigma(x)$.

After N function evaluations, we have at least a constant fraction of the optimal information gain value:

$$F(\mathbf{X_{1:N}}) \geq (1 - 1/e) \max_{n \leq N} F(\mathbf{X_{1:n}}) \tag{15}$$

making this sampling strategy near-optimal. This happens because $F(\mathbf{X_{1:n}})$ satisfies a diminishing returns property called *submodularity*, and the greedy approximation guarantee (15) holds for any submodular function (Nemhauser et al. 1978).

Finally, consequently to the choice of (s', x') according to (13) and (14), the value $y' = f_{s'}(x')$ is observed, at a cost of $c_{s'}$. Then, the dataset $D_{s'}$ is updated as $D_{s'} \leftarrow D_{s'} \cup \{(x', y')\}$ and the overall process is iterated until the cumulated cost does not exceed a prefixed budget. It is anyway possible to include also a stop-criterion related to a maximum number of function evaluations, as typically done in BO. The MISO-AGP algorithm is summarized in Algorithm 1.

Just for completeness, we summarize here the main equations related to fusing GP; for more details, please refer to Ghoreishi and Allaire (2019).

$$\begin{aligned} \mu_{fused}(x) &= \mathrm{k}(x, \mathbf{X}_{1:N_f}) \left[\mathbf{K} + \Sigma(\mathbf{X}_{1:N_f})\right]^{-1} \mu_{wink}(\mathbf{X}_{1:N_f}) \\ \sigma^2_{fused}(x) &= k(x, x) + \\ &\quad -\mathrm{k}(x, \mathbf{X}_{1:N_f}) \left[\mathbf{K} + \Sigma(\mathbf{X}_{1:N_f})\right]^{-1} \mathrm{k}(\mathbf{X}_{1:N_f}, x) \end{aligned} \tag{16}$$

with $\mathbf{K}_{ij} = k(x_i, x_j)\ \forall x_i, x_j \in \mathbf{X}_{1:N_f}$, and $\mu_{wink}(\mathbf{X}_{1:N_f})$ and $\Sigma(\mathbf{X}_{1:N_f}) = diag(\sigma^2_{wink}(x_1), ..., \sigma^2_{wink}(x_{N_f}))$, respectively, the vector of fused means and the diagonal matrix of the fused variances at $\mathbf{X}_{1:N_f}$. These two terms are obtained according to:

$$\mu_{wink}(x) = \frac{\mathbf{e}^T \tilde{\Sigma}(x)^{-1} \boldsymbol{\mu}(x)}{\mathbf{e}^T \tilde{\Sigma}(x)^{-1} \mathbf{e}} \tag{17}$$

$$\sigma^2_{wink}(x) = \frac{1}{\mathbf{e}^T \tilde{\Sigma}(x)^{-1} \mathbf{e}} \tag{18}$$

where $\mathbf{e} = [1, ..., 1]^T$, $\boldsymbol{\mu}(x) = [\mu_1(x), ..., \mu_S(x)]^T$ are the mean values of the S GPs at location x and $\tilde{\Sigma}(x)$ is the covariance matrix between the GPs, whose entry $\tilde{\Sigma}(x)_{ij} = \rho_{ij}(x)\sigma_i(x)\sigma_j(x)$. The value $\sigma^2_i(x)$ is the variance of the GP associated to the i-th source at location x and ρ_{ij} is the correlation coefficient between the deviations of information sources i and j at location x, which is computed as:

$$\rho_{ij}(x) = \frac{\sigma^2_j(x)}{\sigma^2_i(x) + \sigma^2_j(x)} \tilde{\rho}_{ij}(x) + \frac{\sigma^2_i(x)}{\sigma^2_i(x) + \sigma^2_j(x)} \tilde{\rho}_{ji}(x) \tag{19}$$

where $\tilde{\rho}_{ij}$ is computed by *reifying* the i-th GP as:

$$\tilde{\rho}_{ij}(x) = \frac{\sigma_i(x)}{\sqrt{(\mu_i(x) - \mu_j(x))^2 + \sigma^2_i(x)}} \tag{20}$$

Algorithm 1 MISO-AGP algorithm.

set MISO-AGP's parameters: δ and m
$C \leftarrow$ maximum cost
$N \leftarrow$ maximum number of function evaluations
$n \leftarrow 0$
$c \leftarrow 0$
while $c < C$ ***AND*** $n < N$ **do**
  # updating GPs on all the sources
  for $s = 1, ..., S$ **do**
    update $\left(\mu_s(x), \sigma_s(x) | D_s\right)$
  end
  # generating the augmented GP
  $\widehat{D} \leftarrow D_1 \cup \bar{D}$, with $\bar{D}$ as defined in (7)
  update $\left(\widehat{\mu}(x), \widehat{\sigma}(x) | \widehat{D}\right)$
  # computing the augmented best seen
  $\widehat{y}^+ \leftarrow \min_i \widehat{y}_i$, with $\widehat{y}_i : (\widehat{x}_i, \widehat{y}_i) \in \widehat{D}$
  # selecting the next source-point to query
  $(s', x') \leftarrow \arg\max_{\substack{x \in \Omega \\ s=1,...,S}} \alpha_s(x, \widehat{y}^+)$
  with $\alpha_s(x, \widehat{y}^+)$ as defined in (9)
  # applying the correction, if needed
  if $\exists (\tilde{x}, \tilde{y}) \in D_{s'} : ||\tilde{x} - x'|| < \delta$ **then**
    $x' \leftarrow \arg\max_{x \in \Omega} \sigma_1(x)$
    $s' \leftarrow 1$
  end
  # query at (s', x') and observe y'
  $y' \leftarrow f_{s'}(x')$
  # updating the dataset associated to s'
  $D_{s'} \leftarrow D_{s'} \cup \{(x', y')\}$
  # updating cost and function evaluations
  $n \leftarrow n + 1$
  $c \leftarrow c + c_{s'}$
end
 # computing the final augmented best seen
$\widehat{D} = D_1 \cup \bar{D}$
$\widehat{y}^+ \leftarrow \min_{i=1,...,l} \widehat{y}_i$
Result: $\widehat{y}^+$ and $\widehat{x}^+ : (\widehat{x}^+, \widehat{y}^+) \in \widehat{D}$

4 Experimental setting

To validate the proposed MISO-AGP algorithm, we have performed a set of experiments involving both test functions and a real-life application related to AutoML. Experiments are described in the following.

4.1 Test functions

– *Forrester* A one-dimensional function proposed in the MISO setting (Forrester et al. 2007; Bartz-Beielstein et al. 2015). The original function to optimize is given by:

$$f(x) = (6x-2)^2 sin(12x-4) \tag{21}$$

The search space is $\Omega = [0, 1]$, the minimizer is $x^* = 0.7572488$ and the minimum is $f(x^*) = -6.02074$.

The cheaper sources are defined according to:

$$f_s(x) = Af(x) + B(x-0.5) - C \tag{22}$$

In our tests, we have considered two different settings:

1. Two sources available:

$$f_1(x) = f(x) \tag{23}$$

$$f_2(x) = 0.5 f_1(x) + 10(x-0.5) - 5 \tag{24}$$

with costs $c_1 = 1000$ and $c_2 = 1$, as defined in Bartz-Beielstein et al. (2015).

2. Three sources available, adding the third source:

$$f_3(x) = 0.5 f_1(x) + 10(x-0.5) + 5 \tag{25}$$

and costs are $c_1 = 1000$, $c_2 = 1$ and $c_3 = 0.5$.

– *Rosenbrock* A well-known 2-dimensional test function, used as a MISO test in Poloczek et al. (2017). In particular, we have used the first setup of the experiment in Poloczek et al. (2017), but in a noise-free setting:

$$f_1(x) = (1-x_{[1]})^2 + 100(x_{[2]} - x_{[1]}^2)^2 \tag{26}$$

$$f_2(x) = f_1(x) + 0.1 sin(10x_{[1]} + 5x_{[2]}) \tag{27}$$

where $x_{[1]}$ and $x_{[2]}$ are the first and second component of the 2D location x.

Costs of the two sources are $c_1 = 1000$ and $c_2 = 1$. The search space is $\Omega = [-2, 2]^2$, the minimizer is $x^* = (1, 1)$ and the minimum is $f(x^*) = 0$.

4.2 Real-life application: ML

As real-life application, we have considered an AutoML task, more specifically the hyperparameter optimization of an SVM classifier on the "MAGIC Gamma Telescope" dataset, available for free on the UCI Repository.[1] The MAGIC dataset is generated by a Monte Carlo program (CORSIKA Heck et al. 1998), to simulate registration of high-energy gamma particles in a ground-based atmospheric Cherenkov gamma telescope using the imaging technique. The overall dataset consists of 19,020 instances: 12,332 of the class "gamma (signal)," and 6688 of the class "hadron (background)," with each instance represented by 10 continuous features. We have performed a data pre-processing consisting in scaling all the dataset features in [0,1].

As classification learning algorithm, we have selected a C-SVC classifier (Scholkopf and Smola 2001) with a radial basis function (RBF) kernel. The C-SVC's hyperparameters to optimize are the regularization term, C, and γ of the RBF kernel: $k_{RBF}(x, x') = e^{-\gamma \|x-x'\|^2}$.

The misclassification error on 10-fold cross-validation is the objective function to minimize. We have defined two information sources: the first is the misclassification error on 10-fold cross-validation with respect to the entire MAGIC dataset (i.e., $f_1(x) = f(x)$); the second (i.e., $f_2(x)$) is the same performance metric obtained using a smaller portion of the dataset (just 5% obtained via stratified sampling).

Since computational time for querying $f_1(x)$ and $f_2(x)$ depends on the values of C-SVC's hyperparameters, and it is black-box, we have run a sample of 10 hyperparameter configurations on both the two sources and used the average computational times for estimating reference values for c_1 and c_2. More precisely, computational time required by $f_1(x)$ is, on average, 320 times that required by $f_2(x)$. Thus, we set $c_2 = 1$ and, consequently, $c_1 = 320$.

The search space Ω is spanned by the two C-SVC's hyperparameters $C \in [10^{-2}, 10^2]$ and $\gamma \in [10^{-4}, 10^4]$. We adopt a logarithmic scaling of the search space, a usual procedure suggested in AutoML for hyperparameters varying within ranges of this scale.

4.3 Compared approaches

The proposed MISO-AGP has been evaluated according to two different goals. The first one is related to its effectiveness and efficiency with respect to a traditional BO performed on the most expensive source, only. The second one is related to a comparison between using, within the same MISO framework, a fused GP and the proposed AGP. As far as the last consideration is concerned, we want to remark that the MISO-fused GP is not the approach proposed in Lam et al. (2015) or in Ghoreishi and Allaire (2019). More specifically, we use for both MISO-AGP and MISO-fused GP the acquisition function we have proposed—defined in (9) and (14)—while Ghoreishi and

[1] https://archive.ics.uci.edu/ml/datasets/magic+gamma+telescope

Allaire (2019) use a more computational expensive, two-step look ahead function.

Due to the different nature of AGP and fused GP, the final solutions provided by the two approaches are identified by two different mechanisms. For the AGP, as reported in Algorithm 1, the set of inducing locations must be updated at the end of the sequential optimization process and the resulting augmented best seen is returned as final solution. In the case of fused GP, we followed the procedure reported in Ghoreishi and Allaire (2019), consisting in making a further selection according to $\arg\min_{x\in\Omega} \mu_{fused}(x)$ (because we are minimizing), where $\mu_{fused}(x)$ is the fused GP's mean updated at the end of the sequential optimization process. The resulting selected location is then returned as final solution for MISO-fused GP.

Summarizing, the approaches that we have compared in our validation are:

- BO on the most expensive source $f(x)$, only
- MISO-AGP, using the AGP as single model over sources and locations, and (9) and (14) as acquisition function
- MISO-fused GP, using the fused GP as single model over sources and locations, and (9) and (14) as acquisition function.

4.4 Computational setting

All the experiments have been performed on a Microsoft Azure virtual machine, H8 (High Performance Computing family) Standard with 8 vCPUs, 56 GB of memory, Ubuntu 16.04.6 LTS.

The code has been developed in R: all the codes are available upon request from the authors, and an R package is planned to be released soon.

GP modelling, for all the three compared approaches, is performed by using a SE kernel whose hyperparameters are set via MLE. The acquisition function in BO is GP-LCB, while in MISO-AGP and MISO-fused GP we have used (9) along with the correction (14), if required.

As initialization, 2 and 3 initial locations are sampled in Ω via Latin Hypercube Sampling (LHS), respectively for the 1D (Forrester test function) and the 2D (Rosenbrock and C-SVC's hyperparameter optimization) problems. To mitigate the effect of random initialization, 30 different runs of each strategy have been performed for the Forrester and Rosenbrock test problems, and 10 runs for the real-life application. At each run, the three strategies share the same initialization.

As termination criterion, we set a maximum of 30 further function evaluations; for each run, we decided not to set a limit on the cumulated cost but to use the resulting value to compare the different strategies.

5 Results

5.1 Results on test functions

Starting from the *1D Forrester test problem*, the left-hand side of Fig. 5 shows how the values of the best seen—for BO (10) and MISO-fused GP (11)—and augmented best seen—for MISO-AGP (12)—change with respect to the cumulated cost, averaged on the 30 different runs. Solid lines are means and shaded areas represent the standard deviations on the 30 different runs. As expected, both MISO-AGP and MISO-fused GP required, on average, a lower cumulated cost with respect to BO performed only on the most expensive source.

A first relevant consideration is that, in the case of two sources available, the best seen of MISO-fused GP is on average lower than the optimum $f(x^*)$. This inconsistent result is due to the nature of MISO: due to its locally poor approximation, the less expensive source $f_2(x)$ can return values lower than the actual optimum $f_1(x^*)$, as depicted in the right-hand side of Fig. 5. Therefore, the resulting behavior of the best seen implies that MISO-fused GP converged far away from the actual optimizer in many of the 30 runs. On the contrary, the augmented best seen of MISO-AGP resulted on average greater than the actual optimum $f_1(x^*)$, with a small average difference $\widehat{y}^+ - f(x^*)$ and a small standard deviation at last iterations, confirming that it converged close to the true optimizer.

Similar considerations hold for the case with three available sources: the standard deviation related to the best seen of MISO-fused GP implies that some solutions are far away from the global minimizer.

Indeed, these considerations allow us to conclude that looking at the only (augmented) best seen could be misleading in MISO, due to the very different levels of approximation of the sources, also depending on location in Ω. In the right-hand side of Fig. 5, we have depicted the three sources and the locations of the final solutions found over the 30 different runs by each algorithm. As the most relevant result, MISO-AGP on two sources was always able to converge close to the global minimizer, even closer than BO performed on the most expensive source. In the case of three available sources, the final solutions found by MISO-AGP are more spread over the search space Ω, but most of them are anyway close to the global minimizer. On the contrary, MISO-fused GP usually converges far away from the global minimizer.

Results are summarized in Table 1, reporting the distance of the final solutions from the global minimizer x^* (mean and standard deviation on the 30 independent runs), along with the number (and percentage) of them falling into the interval $|x^* - 0.034|$. This corresponds to the smallest region containing all the final solutions for at least one approach, resulting in even smaller than the attraction basin of x^*.

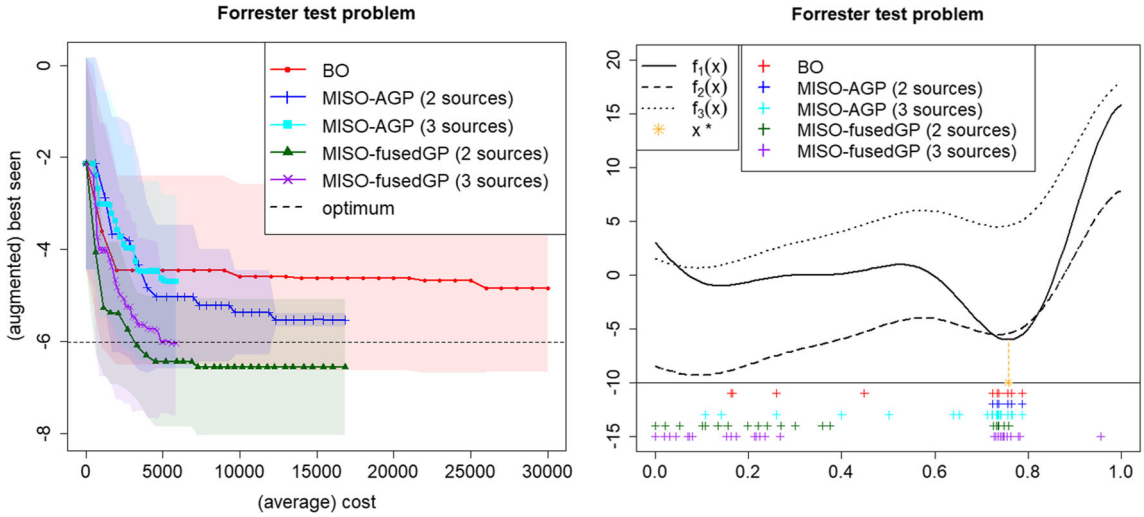

Fig. 5 Results on the 1D Forrester test problem. Left: best seen values with respect to cumulated cost—augmented best seen in the case of MISO-AGP. Both cumulated cost and best seen are averaged over 30 different runs of each algorithm. Right: the three sources of the 1D Forrester test problem with the minimizer and the locations of the final solutions identified by each algorithm in each one of the 30 runs

Friedman's statistical test confirms that the distributions of the distances from x^* are significantly different among the five approaches considered (p-value=0.0368). However, when a pairwise Wilcoxon's test is used, distances from x^* are not significantly different, but in the case of MISO-fused GP with 3 sources compared to BO performed on $f_1(x)$ (p-value=0.005), MISO-AGP with 2 sources (p-value<0.001), and MISO-AGP with 3 sources (p-value=0.004). Thus, MISO-fused GP with 3 sources is the worst approach.

It is important to remark that MISO-fused GP is not the MISO algorithm proposed in Ghoreishi and Allaire (2019): the undesired poor convergence to the actual optimizer might be mitigated by the adoption of their two-step acquisition function, which is anyway more computationally expensive than the GP confidence bound based acquisition function we have proposed.

In any case, the results show that increasing the number of sources from two to three leads to a further reduction, on average, of the cumulated cost, keeping fixed the overall number of function evaluations. This happens because many function evaluations were performed on the cheapest source. Since the selection mechanism is basically driven by the acquisition function and it is the same for both the two MISO approaches, it is reasonable to infer that cost is the most relevant "driver" in selecting the next source to query, more than discrepancy. A possible solution could be to rescale sources' costs in the range [0, 1] by simply dividing each source's cost by the largest one, that is $\hat{c}_s \leftarrow c_s/c_1$, $\forall s = 1, ..., S$. This should lead cost and discrepancy to work in a similar range, giving them the same relevance in the acquisition function.

Table 1 Results on the Forrester test problem: distance from the global minimizer x^*

Approach	Distance from x^* (mean $\pm$ std.dev.)	Final solutions in $\lvert x^* - 0.034\rvert$
BO	0.0927 ± 0.1674	26 [86.67%]
AGP (2 sources)	$\mathbf{0.0309 \pm 0.0079}$	**30** [**100**%]
AGP (3 sources)	0.1065 ± 0.1803	23 [76.67%]
Fused GP (2 sources)	0.3004 ± 0.3056	15 [50.00%]
Fused GP (3 sources)	0.3764 ± 0.3051	11 [36.67%]

Finally, it is important to remark that, given a cumulated cost value, the associated (augmented) best seens obtained using two or three sources are quite similar (Fig. 5). This holds for both MISO-fused GP and MISO-AGP, separately.

The previous considerations are also valid for the *2D Rosenbrock test problem*. Also, in this case, MISO-AGP apparently performed worse than MISO-fused GP, when best seen (11) and augmented best seen (12) are compared (Fig. 6). However, when the distance of the final solutions from the actual global minimizer x^* is considered, the relevant result is that MISO-AGP was able to get significantly closer to x^*.

Also for this test problem we report the distribution of the final solutions found by each algorithm over the 30 different runs (Fig. 7). Although the final solutions found by the two MISO approaches are spread within the search space, most of those identified by MISO-AGP are closer to the actual global minimizer when compared to those found by MISO-fused GP, which instead seems to converge around the location (1.5, 2).

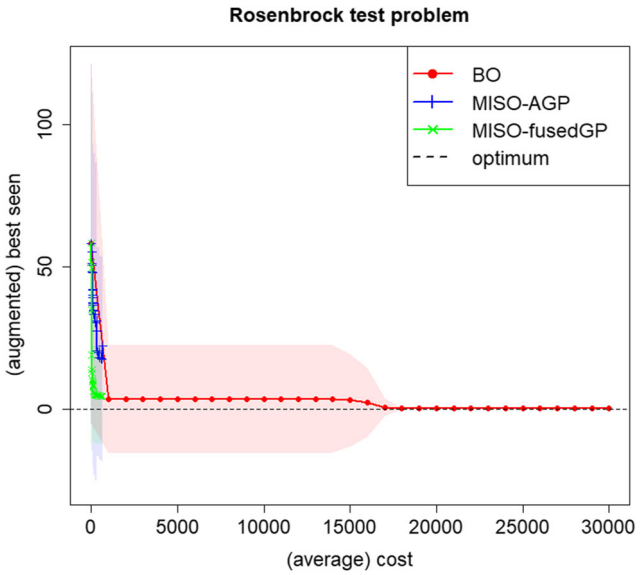

Fig. 6 Results on the 2D Forrester test problem: best seen values with respect to cumulated cost—augmented best seen in the case of MISO-AGP. Both cumulated cost and best seen are averaged over 30 different runs of each algorithm

Results are summarized in Table 2, reporting the distance (mean and standard deviation on the 30 independent runs) of the final solutions, identified by each approach, from the global minimizer x^*. Also, the number (and percentage) of solutions falling into $||x^* - 0.46||$ is reported, that is the smallest circle containing all the final solutions of at least one approach. In this case, BO performed on the most expensive source was able to converge closer to the global minimizer than the other MISO approaches, but of course with a significantly higher cost.

Although only 10 final solutions identified by MISO-AGP (33.33%) are within $||x^* - 0.46||$, they are anyway more than the only 2 identified by MISO-fused GP (6.67%). Enlarging the interval up to $||x^* - 1||$, these figures increase to 17 solutions for MISO-AGP (56.67%) and only 4 solutions for MISO-fused GP (13.33%).

Friedman's test confirms that distance from x^* is significantly different among the three approaches considered (p-value$<$0.001). This is also confirmed by a pairwise Wilcoxon's test: BO solutions are significantly closer to x^* than those of MISO-AGP (p-value=0.001) and MISO-fused GP (p-value$<$0.001). Finally, MISO-AGP solutions are significantly closer to x^* than MISO-fused GP ones (p-value=0.005).

Differently from the Forrester test case, here the shape of $f_1(x)$—and that of its cheap approximation $f_2(x)$—allows to easily converge towards the global minimum. Indeed, in simple cases like the Rosenbrock test problem, using multiple sources could lead to slightly worse solutions than BO performed on the most expensive source: the main advantage is the significant reduction in terms of the overall query cost. On the other hand, in complicated cases, like the Forrester test problem, using multiple sources seems to improve exploration of the search space leading to both better solutions and cost reduction with respect to BO performed on the expensive source. It is important to remark that, due to the black-box nature of the sources, it is impossible to know a priori if we are going to optimize a simple or complicated problem, so the slightly worsening occurring in simple cases could be anyway considered reasonable compared with the significant reduction of the overall query cost.

To provide a comparison with respect to other state of the art MISO approaches, we considered the results reported in Poloczek et al. (2017), for the same test case. This required to perform a further experiment due to some differences in the experimental setup. More precisely, the initial design of this additional experiment consists of 5 randomly selected sample points—instead of 3—for each one of the two sources. As for the other experiments, and coherently with that reported in Poloczek et al. (2017), the random sampling followed an LHS procedure. Although the size of our initial design is the same as that of the experiment considered for the comparison, the sample points within the initial designs are surely different, leading to potentially different results.

Another difference in the experimental setup is related to the performance metric used in Poloczek et al. (2017): they introduce the "gain" over the best initial solution, that is the actual value, computed on the most expensive source, of the solution identified at each iteration minus the best value observed in the initial design. We have therefore computed gain also for the solutions obtained through MISO-AGP on this additional experiment.

In the left-hand side of Figure 1 of Poloczek et al. (2017), it is possible to notice that increasing the cumulated cost, on average, of about 30 units (from 5005 of the initial design up to 5030), the resulting gain achieved by their algorithm is around 27.5, with a standard deviation close to 0. When we consider the same increase in terms of average cumulated cost, the gain achieved by MISO-AGP is 31.09, but with a very large standard deviation (i.e., 40.59). It is important to explain that this high value in standard deviation is mainly due to quite high gain values we have obtained in our experiment: for 9 out of our 30 independent runs (30%) gain was higher than 27.5, with values ranging from 46.40 up to 147.10.

According to a Mann-Whitney U test, the gain obtained by MISO-AGP is significantly higher than that reported in Poloczek et al. (2017) (p-value$<$0.001), at the same cost. More precisely, we compared the gains obtained on

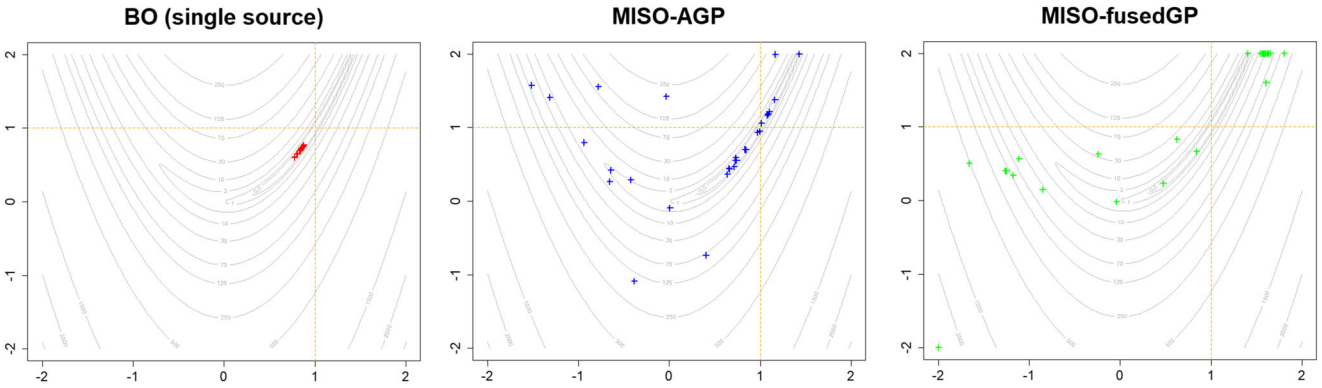

Fig. 7 Distribution of the final solutions found by the three algorithms, on each run, on the 2D Rosenbrock test problem. Level set for the original function are represented (in gray) along with location of the minimizer (whose coordinates are represented by the yellow dashed lines)

30 different runs of MISO-AGP to 100 samples (i.e., the number of runs stated in Poloczek et al. (2017)) drawn form a normal distribution with mean 27.5 and standard deviation 0.5.

5.2 Results on real-life application

Results on the real-life case study are summarized in Fig. 8. On the left-hand side, it is shown how the values of the best seen—for BO (10) and MISO-fused GP (11)—and augmented best seen—for MISO-AGP (12)—change with respect to the cumulated cost, where both best seen and cost values are averaged on the 10 different runs. In this case, both MISO-AGP and MISO-fused GP are significantly more effective and cost-efficient than BO, and almost equivalent between them. Due to the real-life nature of the case study, we do not know neither the optimum nor the optimizer of the problem, so it is difficult to make accurate considerations such as those previously reported for the test problems. To obtain an approximation of the objective function $f_1(x)$, we have used all the function evaluations, independently on the approach, performed on the 10 runs, but only on the most expensive source (i.e., the misclassification error on 10-fold cross-validation on the large dataset for each C-SVC's hyperparameter configuration evaluated). To obtain this approximation, we used a GP with SE kernel. Figure 8, in the middle, reports the distributions of the final solutions identified by the three algorithms, 10 each, and the estimated level sets of $f_1(x)$. Again, some solutions identified by MISO-fused GP are really far from the—approximated—global optimizer, such as the solution in $(-2, 4)$. Finally, on the right-hand side, the 3D reconstruction of the approximated $f(x)$ is reported.

Table 2 Results on the Rosenbrock test problem: distance from the global minimizer $x*$

Approach	Distance from x^* (mean $\pm$ std.dev.)	Final solutions in $\|\|x^* \pm 0.46\|\|$
BO	0.3790 ± 0.0669	30 [100%]
AGP (2 sources)	0.9781 ± 0.7900	10 [33.33%]
Fused GP (2 sources)	1.5434 ± 0.9134	2 [6.67%]

The left-hand side of Fig. 8 is devoted to show and compare the evolution of the best seen for the approaches considered, while the chart in the middle of the same figure depicts the locations of the final optimal solutions of each method. It is important to remark that for selecting the final solutions of MISO-fused GP, we followed the criterion proposed in Ghoreishi and Allaire (2019), consisting in selecting the minimizer of the final fused GP's mean, that is a not-evaluated point. Thus, the locations of the final solutions, for MISO-fused GP, might lie on function levels different from the last associated best seen values.

5.3 Considerations about the role of *m* and δ

MISO-AGP is characterized by two technical parameters, m and δ, respectively used in (7) and (14). We remark that δ is just used to avoid the ill-conditioning problem occurring in GP fitting when the new location to query is too close to one already sampled. To choose an appropriate value for δ, the user has to consider that it could affect the resolution of MISO-AGP: if a location is sampled, closer than δ to the actual global optimizer, then MISO-AGP cannot go closer to that, due to (14).

With respect to m, a more detailed discussion is needed. We remark that increasing m leads to include, as inducing locations of the AGP, a larger number of evaluations performed on the cheap sources. This could degrade the quality of approximation provided by the resulting AGP, as previously shown in Fig. 2. Here, we report an additional experiment for the Forrester test case, extending to $m = 2$ and $m = 3$ the results already reported for MISO-AGP on 3

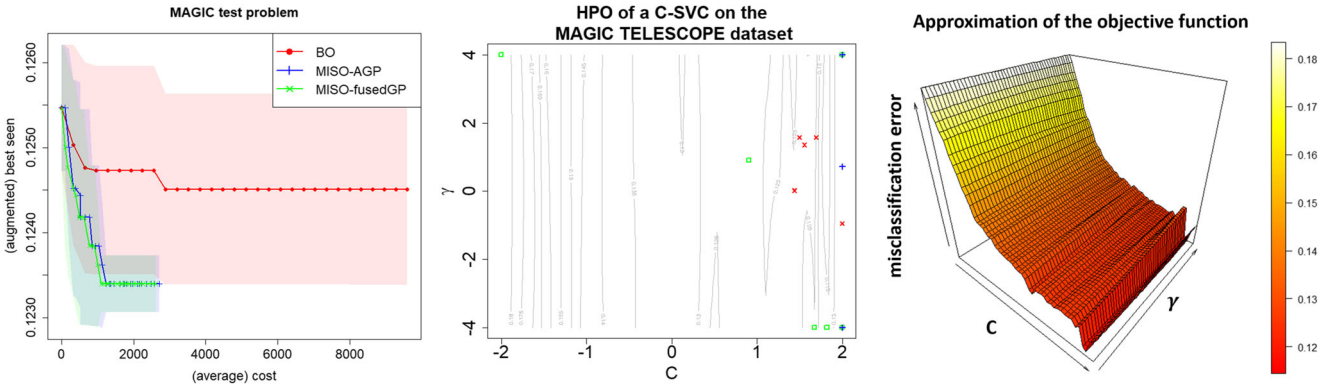

Fig. 8 Results on hyperparameter optimization of a C-SVC classifier on the MAGIC GAMMA TELESCOPE dataset. Left: best seen with respect to the cumulated cost; middle: level sets (estimated) and distribution of the final solutions for the three algorithms (i.e., red crosses for BO, blue plus for MISO-AGP, and green squares for MISO-fused GP); right: 3D reconstruction of the (estimated) original function

sources in order to compare the performance of MISO-AGP with respect to m. All the other experimental settings are the same used in test case 1.

Results are summarized in Table 3, where distances between the final solutions and the global minimizer x^* are reported (as mean and standard deviation) depending on m. We also report the number of final solutions close to the global optimum, more precisely those in the interval $|x^* - 0.034|$ previously defined in Section 5.1 for the same test case. Finally, we also report the overall cost (averaged on the 30 runs).

As the main result, a degradation of performances is observed—as expected—by increasing m. However, this degradation cannot be considered statistically significant (pairwise Wilcoxon test, p-value>=0.1317 for every comparison between two different values of m). More importantly, the reduction in terms of distance and number of solutions close to x^* is not sufficiently counterbalanced by a reduction in terms of the overall costs. Indeed, using MISO-AGP with $m = 1$ allowed reducing the overall cost of around 15,000 s with respect to BO (Fig. 5, left), while using $m = 2$ as well as $m = 3$ led to a further reduction of just around 1200 s.

Finally, Fig. 9 depicts the locations of the final solutions identified by MISO-AGP with respect to the three different values of m.

Table 3 Investigating the role of m in MISO-AGP. Results on the Forrester test case with 3 information sources

m	Distance from x^* (mean $\pm$ std.dev.)	Final solutions in $\lvert x^* \pm 0.034\rvert$	Overall cost [s]
1	0.1065 ± 0.1803	21 [70.00%]	5882.58
2	0.1601 ± 0.2197	18 [60.00%]	4518.32
3	0.1862 ± 0.2328	16 [53.33%]	4617.88

6 Conclusions

The main conclusion is that GP sparsification can be an effective alternative to GP fusion in multi-information source optimization. Advantages confirmed by our experiments are many: (i) MISO-AGP did not outperform MISO-fused GP in terms of value of the best solution, but showed a lower variance in terms of its location, basically due to the AGP's ability to select only reliable function evaluations on less expensive sources, depending on discrepancy and GP's prediction uncertainty; (ii) a significant reduction of the cumulated query cost, with respect to BO on the most expensive sources and aligned with that of MISO-fusedGP; (iii) a lower risk for variance starvation; and (iv) an acquisi-

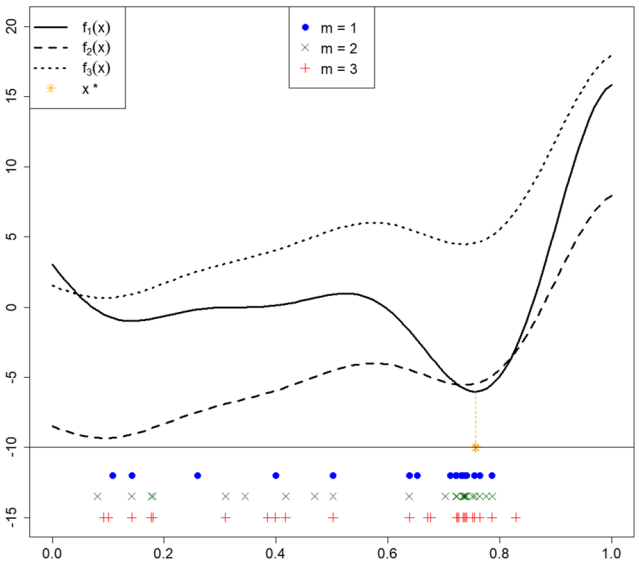

Fig. 9 Locations of the optimal solutions identified by MISO-AGP on 30 different runs, for different values of m

tion function generalizing confidence bound by also considering a pure exploration step (i.e., variance maximization) to avoid premature convergence and ill conditioning.

Moreover, it is important to remark that, contrary to fused GP, the proposed AGP is fitted only on the inducing locations that are less or equal to the function evaluations performed over all the sources. Therefore, at each iteration, fitting an AGP is less computationally expensive than fitting a fused GP.

The computational results compared to other approaches are encouraging: more has to be done to understand whether the gain translates to higher dimensional problems.

Acknowledgements We greatly acknowledge the DEMS Data Science Lab, Department of Economics Management and Statistics (DEMS), University of Milano-Bicocca, for supporting this work by providing computational resources.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Replication of results All the experiments are reproducible since they are related to two test problems and a machine learning application based on an open dataset. All the references are reported in the paper, as well as the algorithm. In any case, the code (developed in R) is available upon request from the authors; an R package is planned to be released soon.

References

Archetti F, Candelieri A (2019) Bayesian optimization and data science. Springer, Berlin

Bartz-Beielstein T, Jung C, Zaefferer M (2015) Uncertainty management using sequential parameter optimization. In: Uncertainty management in simulation-optimization of complex systems. Springer, pp 79–99

Chaudhuri A, Marques AN, Lam R, Willcox KE (2019) Reusing information for multifidelity active learning in reliability-based design optimization. In: AIAA Scitech 2019 Forum, p 1222

Csató L, Opper M (2001) Sparse representation for gaussian process models. In: Advances in neural information processing systems, pp 444–450

Csató L, Opper M (2002) Sparse on-line gaussian processes. Neural Comput 14(3):641–668

Elsken T, Metzen JH, Hutter F (2019) Neural architecture search: a survey. J Mach Learn Res 20(55):1–21

Feldstein A, Lazzara D, Princen N, Willcox K (2019) Multifidelity data fusion: Application to blended-wing-body multidisciplinary analysis under uncertainty. AIAA J, 1–18

Forrester AI, Sóbester A, Keane AJ (2007) Multi-fidelity optimization via surrogate modelling. Proc R Soc A Math Phys Eng Sci 463(2088):3251–3269

Frazier PI (2018) Bayesian optimization. In: Recent advances in optimization and modeling of contemporary problems, INFORMS, pp 255–278

Ghoreishi SF, Allaire D (2019) Multi-information source constrained bayesian optimization. Struct Multidiscip Optim 59(3):977–991

Goel E, Abhilasha E, Goel E, Abhilasha E (2017) Random forest: A review. Int J Adv Res Comput Sci Softw Eng 7(1)

Heck D, Schatz G, Knapp J, Thouw T, Capdevielle J (1998) Corsika: A monte carlo code to simulate extensive air showers. Tech. rep

Hutter F, Kotthoff L, Vanschoren J (2019) Automated machine learning. Springer, Berlin

Kandasamy K, Dasarathy G, Oliva J, Schneider J, Poczos B (2019) Multi-fidelity gaussian process bandit optimisation. J Artif Intell Res 66:151–196

Keerthi S, Chu W (2006) A matching pursuit approach to sparse gaussian process regression. In: Advances in neural information processing systems, pp 643–650

Kennedy MC, O'Hagan A (2000) Predicting the output from a complex computer code when fast approximations are available. Biometrika 87(1):1–13

Klein A, Falkner S, Bartels S, Hennig P, Hutter F (2017) Fast bayesian optimization of machine learning hyperparameters on large datasets. In: Artificial intelligence and statistics, pp 528–536

Lam R, Allaire DL, Willcox KE (2015) Multifidelity optimization using statistical surrogate modeling for non-hierarchical information sources. In: 56th AIAA/ASCE/AHS/ASC Structures, structural dynamics, and materials conference, p 0143

Liu J, Paisley J, Kioumourtzoglou MA, Coull B (2019) Accurate uncertainty estimation and decomposition in ensemble learning. In: advances in neural information processing systems, pp 8950–8961

March A, Willcox K (2012) Provably convergent multifidelity optimization algorithm not requiring high-fidelity derivatives. AIAA J 50(5):1079–1089

Marques A, Lam R, Willcox K (2018) Contour location via entropy reduction leveraging multiple information sources. In: Advances in neural information processing systems, pp 5217–5227

Marques AN, Opgenoord MM, Lam RR, Chaudhuri A, Willcox KE (2020) Multifidelity method for locating aeroelastic flutter boundaries. AIAA J 1–13

Nemhauser GL, Wolsey LA, Fisher ML (1978) An analysis of approximations for maximizing submodular set functions—i. Math Program 14(1):265–294

Pedergnana M, García SG et al (2016) Smart sampling and incremental function learning for very large high dimensional data. Neural Netw 78:75–87

Peherstorfer B, Kramer B, Willcox K (2017) Combining multiple surrogate models to accelerate failure probability estimation with expensive high-fidelity models. J Comput Phys 341:61–75

Poloczek M, Wang J, Frazier P (2017) Multi-information source optimization. In: Advances in Neural Information Processing Systems, pp 4288–4298

Rahimi A, Recht B (2008) Random features for large-scale kernel machines. In: Advances in neural information processing systems, pp 1177–1184

Scholkopf B, Smola AJ (2001) Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press, Cambridge

Schreiter J, Nguyen-Tuong D, Toussaint M (2016) Efficient sparsification for gaussian process regression. Neurocomputing 192:29–37

Seeger M, Williams C, Lawrence N (2003) Fast forward selection to speed up sparse gaussian process regression. Tech. rep

Seeger MW (2008) Bayesian inference and optimal design for the sparse linear model. J Mach Learn Res 9(Apr):759–813

Sen R, Kandasamy K, Shakkottai S (2018) Multi-fidelity black-box optimization with hierarchical partitions. In: International conference on machine learning, pp 4538–4547

Shahriari B, Swersky K, Wang Z, Adams RP, De Freitas N (2015) Taking the human out of the loop: A review of bayesian optimization. Proc IEEE 104(1):148–175

Smola AJ, Bartlett PL (2001) Sparse greedy gaussian process regression. In: Advances in neural information processing systems, pp 619–625

Srinivas N, Krause A, Kakade SM, Seeger MW (2012) Information-theoretic regret bounds for gaussian process optimization in the bandit setting. IEEE Trans Inf Theory 58(5):3250–3265

Swersky K, Snoek J, Adams RP (2013) Multi-task bayesian optimization. In: Advances in neural information processing systems, pp 2004–2012

Thomison WD, Allaire DL (1949) A model reification approach to fusing information from multifidelity information sources. In: 19th AIAA non-deterministic approaches conference

Wahba G (1990) Spline models for observational data, vol 59. SIAM, Philadelphia

Wang Z, Gehring C, Kohli P, Jegelka S (2018) Batched large-scale bayesian optimization in high-dimensional spaces. arXiv:170601445

Williams CK, Rasmussen CE (2006) Gaussian processes for machine learning. MIT Press, Cambridge

Wilson JT, Borovitskiy V, Terenin A, Mostowsky P, Deisenroth MP (2020) Efficiently sampling functions from gaussian process posteriors. arXiv:200209309

Winkler RL (1981) Combining probability distributions from dependent information sources. Manag Sci 27(4):479–488